(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2000－36958
（P2000－36958A）
(43)公開日　平成12年２月２日(2000.2.2)

(51)Int.Cl.'		識別記号	FI		テーマコード'(参考)
H04N	7/24		H04N	7/13	Z
	1/387			1/387	
	1/41			1/41	Z

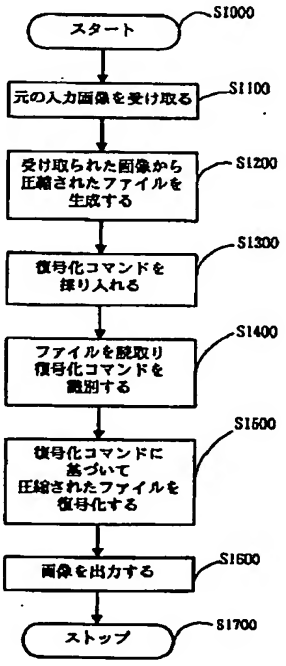
審査請求　未請求　請求項の数３　ＯＬ　外国語出願　（全 40 頁）

(21)出願番号　特願平11－131297
(22)出願日　平成11年５月12日(1999.5.12)
(31)優先権主張番号　０７５９３５
(32)優先日　平成10年５月12日(1998.5.12)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　590000798
ゼロックス　コーポレイション
ＸＥＲＯＸ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　06904－1600　コネティカット州・スタンフォード・ロング　リッチ　ロード・800
(72)発明者　リカルド　エル．デ　ケイロズ
アメリカ合衆国　14534　ニューヨーク州　ピッツフォード　キャンブリック　サークル　26
(74)代理人　100079049
弁理士　中島　淳　（外１名）

(54)【発明の名称】　復号化コマンドを組み込んだ圧縮フレームワーク

(57)【要約】
【課題】　圧縮された画像データファイルに復号化コマンドを採り入れ、圧縮された定義域で実行される処理を補足し、画像形成操作の演算時間を短縮すること。
【解決手段】　圧縮された画像データを処理する方法が、元の画像を表示する電子画像データを受け取り（Ｓ１１００）；該電子画像データから圧縮された画像データファイルを生成し（Ｓ１２００）；復号化コマンドを圧縮された画像データファイルに採り入れ、該復号化コマンドが、原稿画像に関して復号化された画像の外観を変更する命令を有し（Ｓ１３００）；復号化コマンドを識別し（Ｓ１４００）；復号化された画像の処理された電子画像データを形成するために、識別された復号化コマンドに従って圧縮された画像データファイルに含まれる圧縮された画像データを復号化し、該復号化された画像が、処理された原稿画像の変更された態様を表示する（Ｓ１５００）。



1

【特許請求の範囲】

【請求項１】　圧縮された画像データを処理する方法であって、該方法が、

元の画像を表示する電子画像データを受け取るステップを含み、

前記受け取られた電子画像データから圧縮された画像データファイルを生成するステップを含み、

前記圧縮された画像データファイルに復号化コマンドを組み込むステップを含み、該復号化コマンドが、元の画像に関して復号化された画像の外観を変更する命令を有し、

前記復号化コマンドを識別するステップを含み；且つ復号化画像の、処理された電子画像データを形成するために、前記識別された復号化コマンドに従って、前記圧縮された画像データファイルに含まれる圧縮された画像データを復号化するステップを含み、該復号化された画像が、処理された元の画像の変更されたバージョンを表示する、

上記方法。

【請求項２】　圧縮された画像データを処理するシステムであって、該システムが、

元の画像を表示する電子画像データを出力する画像ソースを含み、

前記電子画像データを入力する符号器を含み、該符号器が、

前記電子画像データから圧縮された画像データファイルを生成する圧縮装置を有し、及び前記圧縮された画像データファイルに復号化コマンドを組み込む追加装置を有し、該復号化コマンドが、前記元の画像に関して復号化された画像の外観を変更する命令を有し；復号化された画像データを形成するために、前記圧縮された画像データを復号化する復号器を含み、該復号器が、

前記復号化コマンドを処理するコマンドプロセッサと、復号化デバイスとを有し、

ここで、前記復号器が、前記復号化コマンドに基づいて、前記圧縮された画像データファイルに含まれた圧縮された画像データを復号化し、再構成された画像を形成し、該再構成された画像が、元の画像の変更されたバージョンを表示する、

上記システム。

【請求項３】　圧縮された画像データファイルのためのデータ構造であって、該構造が、

ヘッダ、及び圧縮されたデータ部を含み、

ここで、前記ヘッダが、

識別部、

画像情報部、

測光部、

コード部、及びコードブック画定部、

のうちの少なくとも１つを含む、上記構造。

【発明の詳細な説明】

2

【０００１】

【発明の属する技術分野】本発明は、全般に、圧縮されたデジタル画像の処理に関する。更に詳細には、本発明は、画像形成操作の演算時間を短縮するために、復号化コマンドを圧縮された画像データファイルに組み込み、圧縮された定義域で実行される処理を補足する方法及び装置に関する。

【０００２】

【従来の技術及び発明が解決しようとする課題】データの圧縮は、データを用いた実用的アプリケーションのための、データが過剰に存在するデータ処理プロセスにおいて必要とされる。通常、圧縮は、伝送時間又は必要とされる帯域幅を短縮又は縮小するために、通信リンクにおいて用いられる。同様に、圧縮は、プリントされるドキュメントの「ページ」が事前照合メモリに一時的に記憶されるデジタルプリンタ及び複写機を含む画像記憶システムで、好まれる。画像データが記憶される媒体空間の量は、圧縮によって実質的に軽減され得る。概して、走査された画像、すなわち、プリントされたドキュメントの電子表示は大抵大きいため、圧縮が望まれる対象物となる。

【０００３】本発明は、元の画像の外観を変更する画像形成操作の演算時間の短縮に関する。特に、本発明は、圧縮された画像データファイルに復号化コマンドを採り入れる。これらのコマンドは、圧縮解除装置（デコンプレッサ）に伝達される命令及び圧縮された定義域で実行される補足処理である。コマンドは、可能性として元の画像のパラメータと共に、元の画像に関して復号化された画像の外観を変更するよう、圧縮解除装置に指示する。次に、圧縮解除装置は、データを圧縮解除すると共に、いくつかの処理ステップをデータに対して実行する。これは、画像操作の演算時間の短縮をもたらす。

【０００４】

【課題を解決するための手段】本発明の１つの態様は、圧縮された画像バッファの頻繁な利用、すなわち費用のかかる圧縮解除及び再圧縮操作を行わずに画像データを処理することができないこと、による基本的な問題を扱う。定義域における処理をよりアクセス可能にするために、本発明に従った圧縮フォーマット／方法論は、所定量のデータが画像内の所定数の画素に対応する方法によって、圧縮されたデータを生成するのが好ましい。この条件が維持される場合、圧縮されたデータの画像領域は、容易に識別、置換、又はクロップ（作成）される。本発明の１つの好ましい実施形態は、圧縮された定義域でいくつかの画像処理操作が容易に適用されることを可能にする、復号化コマンドのための構文及び画像を圧縮する方法を含む。

【０００５】本発明のシステム及び方法の好ましい実施形態は、ベクトル量子化（ＶＱ）などの圧縮体系に特によく作用する。ベクトルの量子化において、画素の群

は、コードベクトルとして群を表示することにより、同時に量子化される。画像データは、まず、数セットのベクトルに処理される。次に、量子化されるデータに最もよく適合するコードブックが生成される。各入力ベクトルは、次に、最も近いコードワードに量子化される。圧縮は、コードワードの指標のみを伝送することにより、達成される。画像は、レシーバにおいて、テーブルルックアップ手順を用いて再構成される。

【０００６】本発明のシステム及び方法の好ましい実施形態において、符号器と復号器は、コードブックを共用する。本実施形態において、コードワードは、補助データの形態を取り、圧縮された画像データが該補助データに基づいて生成される。補助データの符号化指標が伝送され、符号器は、これらの符号化指標から１セットの画素をルックアップする。従って、圧縮された画像を処理するために、復号化コマンドが圧縮された画像データファイルに採り入れられうる。復号化コマンドを受け取る圧縮解除装置は、コマンドを理解し、コードブックエントリ及び／又は圧縮された画像データにそれら（コマンド）を実行する。補助データは、復号化コマンドに基づいて処理される。圧縮された画像データの復号化は、処理された補助データに基づく。コードブックエントリは、復号化プロセスとは無関係であるため、指標とコードブックエントリの対応が維持される限り、画像は通常通りに圧縮解除される。

【０００７】よって、本発明の方法及びシステムは、ＴＲＣマッピング、ビットデプス(bit-depth)の変換、リサイジング、回転、ミラリング(mirroring)、転置、ハーフトーニング、クロッピング、ペースティング及びマージングを含む、あらゆる数の操作を圧縮されたデータに適用することができる。本発明のシステムは、画像を圧縮及び／又は圧縮解除することができるデータ又は画像処理システムを含みうる。本明細書に述べられる方法及びシステムは、それらが画像形成操作の演算時間の短縮をもたらすため、有用である。

【０００８】本発明のこれら及び他の、特徴と利点は、以下の好ましい実施形態の詳細の説明において述べられる、すなわち、それらは該実施形態の詳細の説明から明白となる。本発明の好ましい実施形態は、その詳細を以下の図を参照して説明されるであろう。

【０００９】

【発明の実施の形態】図１は、本発明に従った符号化及び復号化システム１００の汎用化された機能ブロック図を示す。システム１００は、電子画像データを生成するのに好適であるスキャナ、デジタル複写機又はファクシミリマシンデバイス、又はネットワークのクライアントやサーバなどの電子画像データを記憶及び／又は伝送するのに好適なデバイスなどの多数の種々の画像ソースのうちの何れか１つであってもよい画像ソース１１０を備える。画像ソース１１０から出力される電子画像データは、信号ライン１２０を通って符号器４００に付与される。符号器４００は、本発明に従った画像処理及び圧縮操作を実行することが可能であると共に、追加の機能性を設けるのが好ましい。図１は、符号器４００をＶＱ符号器として示すが、無損失又は損失のあるあらゆるタイプの画像の圧縮が、本発明に従って、符号器４００によって実行されうる。このような場合、符号器４００の特定のエレメントは図１に示されるものとは異なってもよい。つまり、本発明は、厳密に言えば、ＶＱ符号化に限定されず、画像の符号化又は圧縮のあらゆる知られているタイプ又は後に発明されるタイプを含む。

【００１０】図１に示される１つの好ましい実施形態において、符号器４００は、電子画像データを複数の$M_1 \times M_2$ブロック又はセグメントに分ける画像ブロッキング部４１０を含む。ある実施形態において、ブロッキング操作は、１ブロック又はそれ以上の$M_1 \times M_2$ブロックのデータを有するデータの、入力ドキュメントから符号器４００の$M_1 \times M_2$データバッファへの転送を可能とする、ウィンドウ化又はクロッピング回路によって達成されてもよい。データバッファに記憶されると、次に、$M_1 \times M_2$ブロックの画像データは、信号ライン４２０を通って送信され、ＶＱ符号器４３０によって操作される。画像データをブロック内に圧縮するために、ＶＱ符号器４３０において、データがベクトル量子化技術と関連した様々な圧縮操作を用いて操作されてもよい。

【００１１】圧縮された場合、$M_1 \times M_2$ブロックの画像データの各々は、次に、信号ライン３２０を通ってチャネル又は記憶装置３００に伝送されるのが好ましい。チャネル又は記憶装置３００は、圧縮された画像データを復号器５００に伝送するチャネル装置か、又は圧縮された画像データを圧縮解除する必要性が生じるまで、圧縮された画像を無制限に記憶するための記憶装置に伝送するためのチャネル装置かの、何れか又は両方であり得る。チャネル装置は、圧縮された画像データを本発明に従った符号器４００を満たす第１の装置から、物理的に遠隔の本発明に従った復号器５００に、信号ライン３４０を通って伝送するためのあらゆる知られている構造又は装置でありうる。従って、チャネル装置は、公共の交換電話ネットワーク、ローカル又は広域ネットワーク、イントラネット、インターネット、ワイヤレス伝送チャネルなど、又はあらゆる他の分散形ネットワークであってもよい。

【００１２】同様に、記憶装置は、圧縮された画像データを無制限に記憶する、あらゆる知られている構造又は装置、すなわち、ＲＡＭ、ハードドライブ及びディスク、フロッピードライブ及びディスク、フラッシュメモリなど、であってもよい。更に、記憶装置は、符号器４００及び／又は復号器５００から物理的に離れ、上述のチャネル装置により通信可能であってもよい。

【００１３】チャネル又は記憶装置３００の出力は、復

号器５００によって復号化されるデータのコード化された、又は圧縮されたユニットである。上述の符号器４００と同様に、図１は復号器５００をＶＱ復号器として示すが、復号化される圧縮された画像データファイルを生成するために用いられる圧縮のタイプが一致する限り、無損失又は損失のあるあらゆるタイプの圧縮解除が、本発明に従って復号器５００によって実行されることが理解されるべきである。このような場合において、復号器５００の特定のエレメントは、図１に示されるものとは異なってもよい。つまり、本発明は、ＶＱ復号化に限定されず、厳密に言えば、画像の復号化又は圧縮解除のあらゆる知られているタイプ又は後に発明されるタイプを含む。

【００１４】図１に示される好ましい実施形態において、復号器５００は、信号ライン３４０を介してチャネル又は記憶装置３００から圧縮された$M_1 \times M_2$ブロックのデータを受け取るＶＱ復号器５３０と、信号ライン５２０を介してＶＱ復号器５３０から復号化された画像を受け取る画像プロセッサ５１０とを含む。図１で、復号器５００は、符号器４００とは物理的に分離して示されるが、復号器５００及び符合器４００は、単一の物理装置の異なる態様であってもよいことが理解されるべきである。

【００１５】復号器５００の画像プロセッサ５１０から出力される再構成された画像は、信号ライン２２０を通って出力装置２００に伝送されうる。出力装置２００は、本発明に従って処理された画像を出力又は記憶することが可能なあらゆる装置であり得ることが理解されるべきである。

【００１６】図２は、従来のＶＱ符号器を示す。従来のＶＱ符号器４３０において、$M_1 \times M_2$ブロックの画像データはベクトル量子化器４３２に入力される。ベクトル量子化器４３２において、入力された画像データのブロックは、符号化コードブック４３４に記憶されるエントリと比較される。入力されたブロックに最も近いコードブック４３４中のエントリが選択される。従って、量子化器ベクトル４３２は、入力された画像データのブロックのために選択されたコードワードエントリのコードブックの指標のみを出力する。この指標、すなわちコードワードは、選択的に、エントロピー符号化、例えば、算術符号化、ハフマン符号化、ＬＺＷ符号化など、を用いて、更に符号化されることができる。

【００１７】図３は、従来のＶＱ復号器５３０を示す。従来のＶＱ復号器５３０において、Ｂビットのコードワードは、テーブルルックアップデバイス５３２に入力され、指標が復号化コードブック５３４に入力される。コードブック５３４中の指標に対応するピクセルパターンエントリは、復号化された画像ブロックとしてテーブルルックアップデバイス５３２から出力される。

【００１８】実際的なＶＱの実施の例として、階層ＶＱ（ＨＶＱ）が挙げられる。図４は、最終コードワードに量子化された入力画像の、０番目のレベルの８つの画像画素（レベル０）を含む、ＨＶＱ実施の一例を示す。該入力画像の０番目のレベルの８つの画像画素は、ベクトル量子化の１段階を経て、入力画像の第１のレベルの４つのコードワードを形成する。該入力画像の第１のレベルの４つのコードワードは、入力画像の第２のレベルの２つの新しいコードワード（レベル２）に量子化される。該入力画像の第２のレベルの２つの新しいコードワードは、最終的に、入力画像の第３のレベルのコードワード（レベル３）に量子化される。この例において、７つのテーブルルックアップが用いられる。ＨＶＱ符号器４３０は、連続したテーブルルックアップを実行しなければならないが、最終のコードワードが復号化された画像の$M_1 \times M_2$ブロックを完全に特定するため、ＨＶＱ復号器５３０は、その逆の径路に従う必要がない。

【００１９】図５は、各ステップにおいて、２つのＮビットの記号が、ルックアップテーブル（ＬＵＴ）４３６を介し、固定Ｂビットコードワードを用いて符号化されるＨＶＱ実施の一例を示す。コードワードが指定された後、コードワードは伝送される。図６に示されるように、復号器は、ルックアップテーブル（ＬＵＴ）５３６を介して、受け取られたコードワードに基づいて２つのＮビット記号を再構成することができる。

【００２０】ＨＶＱ実施では、各ステップで、記号の数を２で割り、２つの記号は、出力される１つの符号化された記号に結合される。各ステップにおいて、記号の数は割られる。従って、Ｓステップの後、記号の数は2^Sで、単一の記号まで割られる。事実上、減らされた（割られた）数のビットが伝送されるため、圧縮は達成される。目的は、プロセスによって受けるひずみを、優れたデザイン技術で最小化することである。利点は、２Ｎから２Ｂビットへの変換のためのテーブルのステージのみが設けられることが必要とされることによる計算の単純化である。

【００２１】階層ベクトル量子化などの圧縮体系は、共に、符号化及び復号化に対して非常に高速である。更に、本発明の実施形態において表されるように、それらは、復号化コマンドの利用のためにも非常に実践的である。

【００２２】一般に、従来の圧縮された画像データファイルは、圧縮された画像データ、及び元の画像パラメータを提供する追加の情報を備えることが理解されるべきである。復号器は、これらの元の画像パラメータが圧縮された画像データを構成することを必要とする。つまり、この追加の情報なしで、復号器が、復号化された画像データを構成する方法を決定し、画像を形成することは、不可能であろう。この追加の情報は、図１０及び１１に示される１つ又はそれ以上の画像情報部７２０及び測光部７３０を含むのが一般的である。この追加の情報

は、図１０及び１１に示されるように、一般に、圧縮された画像データ７６０と結合されて圧縮された画像データファイル７００を形成するヘッダ７７０に設けられる。

【００２３】本発明に従った復号化コマンドは、復号化コマンドが圧縮された画像データファイルに付加される際、ヘッダに付与された従来の付加的情報を圧縮された画像データファイルと結合して、該復号化コマンドは、圧縮された画像データを圧縮解除することから形成される出力画像が、圧縮された画像データが作成された元の画像と異なることを許容するような、あらゆる情報を含むことが理解されるべきである。出力画像が、以下に限定されるわけではないが、指向方向、サイズ、スケーリング、アスペクト比、ビットデプス、及びあらゆるこれらの組合せ、並びに元の画像に成されうる他の変形を含むあらゆる点で元の画像と異なり得ることが更に理解されるべきである。

【００２４】なお、圧縮された画像データファイルが符号器４００によって作り出される時から、圧縮された画像データファイルが復号器５００によって復号化される時の間の何時でも、復号化コマンドが圧縮された画像データファイルに付加されることが可能なことが更に理解されるべきである。

【００２５】ヘッダに付加された本発明に従った復号化コマンドにより、圧縮されたデータファイルは、圧縮解除された画像データがどのように解釈されるか及び／又は復号化コードブック５３４のコードブックエントリによってどの機能が実行されるか又は復号器５００により該コードブックエントリにどの機能が実行されるかが特定される。コードブック５３４は、画像に比べて、大抵小さいため、コードブック５３４上への操作は計算上の手間がかからないのが一般的である。同様に、圧縮解除されたデータがどのように解釈されるかを変更するのにも計算上の手間がかからないのが一般的である。復号器５００が、入力されたデータファイルを読取り、復号化コマンドを識別すると、復号器５００は入力されたデータファイルのフィールドを変更し、例えば、コードブックへの操作などにより、復号化コマンドに従って、圧縮された画像データを復号化する。全てのコードブックエントリに、１ブロックの$M_1 \times M_2$画素が処理される。

【００２６】図７に示されるように、本発明の１つの実施形態に従って、符号器４００で、ベクトル量子化器４３２によって補助データとして出力されるコードワードは、ライン４３６を通って復号化コマンド追加器４３８に入力され、ここで、復号化コマンドは、圧縮された画像データファイルのヘッダを形成するために、圧縮された画像データに追加された追加の情報に採り入れられ、追加される、すなわち更に一般的には、組み込まれる。

【００２７】本発明の１つの好ましい実施形態において、復号化コマンド追加器は、図１０及び１１に示されるように、ヘッダ７７０のコード（ＣＯＤＥ）部７４０に変更を付加するか又は変更を成す。コード部は、ビットデプスフィールド、ビット幅フィールドなどの、１つ又はそれ以上のデータフィールドを含む。

【００２８】これらのフィールドのそれぞれのデータは、本来、元の画像データ及び／又は元の復号化体系、並びに指標を生成するために用いられるコードブックに基づいてセットされる。従って、例えば、８ビットバイトマップ又は連続階調画像データのための、ヘッダのビットデプスのフィールドは、当初は「８」にセットされる。同様に、画像データが４のブロック幅と４のブロック高を用いて符号化される場合、ヘッダのブロック幅及びブロック高フィールドは、当初は「４」にセットされる。同様に、指標を生成するために特定のコードブックが用いられる場合、コードブックフィールドは、ヘッダに明示的に示されているコードブックを示すラベルか、又はコードブックが復号器に既に知られている場合は、圧縮された画像データを復号化するために用いられるコードブックを識別するラベルかの何れかにセットされる。

【００２９】次に、ヘッダのコード（ＣＯＤＥ）部７４０における１つ又はそれ以上のデータフィールドの値を、画像データ及び／又は符号化体系又はコードブックから決定される元の値から変更することにより、ヘッダのコード（ＣＯＤＥ）部７４０のデータフィールドの変更された値に基づいて圧縮された画像データを復号化する際、変更され、かつ圧縮解除された画像を形成するために、復号器はコードワードを変更し、変更されたコードワードに基づいて圧縮された画像データを復号化する。

【００３０】本発明に従って、図８に示されるように、復号器５００において、コードブックの指標及び追加された復号化コマンドは、コマンドプロセッサ５３８に送られ、ここで、コードワードが読取られ、復号化コマンドが識別される。ヘッダにおいて変更されるフィールドは、例えば、ビットデプス（ＢＤ）、すなわち再構成された画像において所望される画素毎のビットの量；ブロック幅（ＢＷ）、すなわち再構成されたブロックの画素の幅；ブロック高（ＢＨ）、すなわち再構成されたブロックの画素の高さ；などを含みうる。次に、コードワードが、復号化コマンドに基づいて処理される。コマンドプロセッサ５３８からの出力は、信号ライン５３６を介してテーブルルックアップデバイス５３２に送られる。次に、テーブルルックアップデバイス５３２は、コードブック５３４上に操作を開始する。例えば、ヘッダに明示的に設けられるか、又はヘッダに名前で識別されるかの何れかである、新しいコードブックデータが、信号ライン５３５を介してコマンドプロセッサ５３８からコードブック５３４に提供され、先に記憶されたコードブックデータを置き換える。圧縮された画像データは、処理

されたコードワードに基づいてしかるべく復号化される。

【００３１】上記に示されるように、復号化コマンド追加器４３８は、ＶＱ符号器４３０に設けられる代わりに、又はＶＱ符号器４３０に加えてＶＱ復号器５３０に、設けられてもよいことが理解されるべきである。この場合、復号化コマンド追加器４３８が復号器５００に設けられると、復号化コマンド追加器４３８は、受け取られた圧縮された画像データファイルを信号ライン５２０を介して入力する。この受け取られた圧縮された画像データファイルは、該データファイルに追加された復号化コマンドを既に有するか又は有さないこともある。ＶＱ復号器５３０の復号化コマンド追加器４３８は、受け取られた圧縮された画像データファイルを入力し、圧縮された画像データファイルに復号化コマンドを付加するか、又はそのヘッダに既存する復号化コマンドを変更して、元の画像とは異なる所望の出力画像を所望の方法で得る。上述のように、校正された圧縮された画像データファイルは、次に、復号化コマンド追加器４３８によって、コマンドプロセッサ５３８へ出力される。同様に、復号化コマンドは、符号器４００による出力から復号器５００による入力の間の何れの時期においても代替的に追加され得る。

【００３２】復号化コマンドのセットは、特定のサイズに収まるようにブロックをスケーリング（拡大縮小）し、ブロックエントリを再配列し、且つ画素毎の操作を実行するための命令を有するのが好ましい。特定のサイズに収めるためのブロックのスケーリングは、例えば、ビットデプスを変更する操作、元はM_1画素であったブロック幅を変更する操作、又は、元はM_2画素であったブロック高を変更する操作などを含む。ブロックエントリの再配列は、mnの数を有するベクトルｖを用いることにより達成される。元のブロックが、固有の走査順序、すなわち左から右又は上部から下部で、ベクトルｘに配列される場合、ブロックエントリの再配列は、ベクトルｙに帰着する。ｘからｙへのマッピングは、下記式によって制御される。

【００３３】

$$y[k] = x[v[k]] \quad k = 1, \ldots, mn$$

【００３４】追加される復号化コマンドに、入力が復号化コードブックの１ブロックのエントリであるプログラムコードなどの、より複雑なコマンドが含まれ得ることが理解されるべきである。例えば、コマンドは、ＪＡＶＡ™コード、又は他の何れの所定の構文法にも付与されることができる。

【００３５】図９は、本発明に従った画像の符号化、処理及び展開方法の概要を記述するフローチャートである。制御は、ステップＳ１０００で開始され、元の画像の電子画像データが入力されるステップＳ１１００へと続く。次に、ステップＳ１２００において、圧縮された画像データが電子画像データから生成される。続いて、ステップＳ１３００において、復号化コマンドが、圧縮された画像データに採り入れられる。次に、制御はステップＳ１４００へと続く。

【００３６】ステップＳ１４００において、データが読取られ、復号化コマンドが識別される。次に、ステップＳ１５００において、復号化コマンドが識別されると、復号器は、識別された復号化コマンドに基づいて圧縮された画像データを復号化することを開始する。このステップは、例えば、コードブック上の操作、を含む。次に、ステップＳ１６００において、生成された画像が出力される。続いて、ステップＳ１７００において、処理はストップする。

【００３７】図１０及び１１は、ステップＳ１５００においてコードブックがどのように操作されるかを更に詳細にわたって示す。図１０は、例示のヘッダ７７０と圧縮された画像データ７６０を含む、例示の圧縮された画像データファイル７００を示す。圧縮された画像データファイル７００は、４画素×４画素のブロックでＶＱ符号化を用いて圧縮された単色の９６０画素×１０２４画素の元の画像から生成され、そのため、各４画素×４画素ブロックはコードワードにマッピングされる。特に、図１０は様々な復号化コマンドを例示するが、これらの復号化コマンドは、元の画像と比較して復号化された画像に対する何れの変形ももたらさない。

【００３８】ヘッダ７００のＩＤ部７１０は、復号器５００に対するフォーマットを識別する。先に論じられたように、ヘッダ７００の画像情報（ＩＭＡＧＥＩＮＦＯ）部７２０は、復号器５００への追加の情報として元の画像のパラメータを提供する。特に、図１０で示されるように、画像情報（ＩＭＡＧＥＩＮＦＯ）部７１０は、元の画像が、２４０×２５６画素の配列で、単色プレーンのみが存在すること、すなわち画像は単色（白黒）であること、を示唆する。先に論じられたように、ヘッダ７００の測光（ＰＨＯＴＯＭＥＴＲＹ）部７３０は、復号器５００に、圧縮された画像データを復号化する際に使用される、元の測光又は色空間を提供する。図１０に示されるように、復号器５００によって用いられる元の測光、又は色空間は、グレー直線測光である。

【００３９】ヘッダ７００のコード（ＣＯＤＥ）部７４０は、元の画像に関して出力画像を変更させ得る復号化コマンドを付与する。先に論じられたように、コード（ＣＯＤＥ）部７４０は、出力コードブック、圧縮、出力ビットデプス、出力ブロック高、出力ブロック幅、出力ＴＲＣマッピング、及び出力走査順序などのフィールドを含む。これらのフィールドの値を、画像を圧縮する際に用いられた値から変更することにより、復号器５００は、これらのフィールドの変更されたもの（箇所）に基づいて、変更された値を有するよう、出力画素を変換するために、伝送されたコードワード又はコードワード

の指標に対応する出力画素のセットを処理する。これについて、以下で図１１に関し、更に詳細が説明されるであろう。

【００４０】図１０に示されるように、特に、コード（ＣＯＤＥ）部７４０は、出力画像が、自然な走査順序で線形マッピングを経て、１画素につき８ビットを有する４画素×４画素のブロックとして復号化されるべきことを示唆する。コードブックは、特定のコードブックラベル、すなわち「Default_１０２４」にセットされる。上述のように、このラベルは、ヘッダ７７０のコードブック部７５０に明示的に定められるコードブックか、又は、圧縮されたデータ７６０を復号化するために用いられようとしている一般の復号器５００又は特定の復号器５００に利用可能であることが知られるコードブックかの、何れかを示すことができる。セットされる特定のコードブックがない場合、所定のコードブックが用いられる。コードワードアレイは、全く圧縮されない。

【００４１】図１０に示されるように、ヘッダのコードブック部７５０は、空でなければ、事実上のコードブックエントリを示す。圧縮された画像データファイル７００の圧縮されたデータ部７６０は、符号化された画像データの６１４４０（２４０×２５６）ブロックの実際のコードブックエントリの指標を示す。

【００４２】コード（ＣＯＤＥ）部７４０'に設けられるコードブックラベルは、元の画像データを圧縮するために用いられるコードブックとは異なったコードブックを示しうることが理解されるべきである。更に、元の画像データを符号化するために用いられるコードブックは、コード部７４０のコードブックラベルによって識別されるコードブックと同じラベルを使用することができるが、２つのコードブックは同じでなくてもよい。従って、コードブック部７５０は、名目上、元の画像データを圧縮又は符号化するために用いられるコードブックと同様のラベルを有するが、元の画像を圧縮又は符号化するために用いられるコードブックとは異なるコードブックを明示的に画定することができる。同様に、コード部７４０のコードブックラベルが単に、既に復号器５００に利用可能なコードブックを識別するとしてもなお、その識別されたコードブックは、元の画像データを圧縮又は符号化するために用いられるコードブックとは異なりうる。

【００４３】図１１は、圧縮されたデータファイルを復号化すると、出力画像の外観を変更するヘッダ７７０'を有する例示的な圧縮された画像データファイル７００'である。コード（ＣＯＤＥ）部７４０'のヘッダフィールドのいくつかを変更することにより、画像データブロックは、復号器５００によってしかるべく処理され、変更された、復号化された画像を付与する。図１１において、ビットデプスフィールドは図１０に示される「８」から「２」に変更され、ビット高フィールドは図１０に示される「４」から「２」に変更され、また、ビット幅フィールドは図１０に示される「４」から「２」に変更される。従って、ＶＱ符号器４００でコードワードを生成する際に用いられるこれらのフィールドの値と比較して、これらの値は、入力画像の２分の１（２／４）の幅、すなわち水平方向の広がりと、入力画像の２分の１（２／４）の高さ、すなわち垂直方向の広がりと、入力画像の４分の１（２／８）のデプス、すなわちグレースケールレベルとを有する出力画像をもたらす。

【００４４】図１１において、ＴＲＣマッピングフィールドと走査順序フィールドもまた、図１０と比較すると変更されている。特に、図１１に示されるコード（ＣＯＤＥ）部７４０'において、ＴＲＣフィールドの変更により、復号化された画素のグレースケール値が図１０に示される当初のグレースケール値から変更される。図１０において、出力画像の各グレースケール値は、ＴＲＣフィールドに示される０から始まるリストにおける値の位置に対応する。図１１において、１２８より下の位置は、それらの位置の値から０に向って移動される値を有する。対照的に、１２７より上の位置は、それらの位置の値から２５６へと移動するそれらの値を有する。これらの値のシフトの作用は、明と暗の画素のコントラストを上昇させることである。

【００４５】加えて、図１１に示されるコード（ＣＯＤＥ）部７４０'における、走査フィールドの変更により、ブロックの配置を、走査ラインにおけるそれら（ブロック）の位置に従って、図１０に示される走査フィールドで表示される入力順序から、図１１に示される走査フィールドで定められる出力順序に変更される。

【００４６】他のフィールドが変更されてもよく、また、図１１において、ビットデプス、ビット高、ビット幅、ＴＲＣマッピング及び走査フィールドの全てが変更されるとして示されるが、それぞれが独立して変更されることができると理解されるべきである。

【００４７】また、追加の復号化コマンドが、図１０及び１１に示されるヘッダ７７０及び７７０'に付加されることができる。これらの追加の復号化コマンドは、コード（ＣＯＤＥ）部７４０の復号化コマンド及び／又はコードブック部７５０に成された変更に付加されるか又はそれらの代わりとなりうる。

【００４８】これらの追加の復号化コマンドは、元の画像パラメータを変更するために用いられる。元の画像パラメータに対するこれらの変更は、画像情報（ＩＭＡＧＥＩＮＦＯ）部７２０及び／又は測光(Photometry)部７３０で当初示された画像パラメータを変更することよりもむしろ、画像情報（ＩＭＡＧＥＩＮＦＯ）部７２０及び／又は測光部７３０に復号化された画像パラメータを付加することにより実行される。つまり、これらの追加の復号化コマンドを用いるために、コマンドプロセッサ５３８は、元の画像パラメータと、所望の出力画像用の

所望の画像パラメータの復号化コマンドの両方を有する必要がある。コマンドプロセッサ５３８は、圧縮解除された画像データが所望の画像パラメータを有するように、圧縮されたデータ部７６０において圧縮された画像データがどのように圧縮解除されるべきかを決定するために、元の画像パラメータと所望の画像パラメータの復号化コマンドを用いる。

【００４９】図１２～１７は、本発明に従って追加された復号化コマンドを用いて実行されうる処理操作の例を示す。図１２は、画素毎の操作として階調再現曲線（ＴＲＣ）を示す。この操作において、全ての入力画素値は、曲線上のもう１つの画素値にマッピングされる。図１２の部分（ａ）において、出力画像のコントラストは、入力画像に相関して低下される。図１２の部分（ｂ）において、出力画像の明るさが増加される。図１０～１１の例示のヘッダを用いて画像のコントラスト又は明るさを高めるための、ＴＲＣの画素毎の操作は、図１０に示されるヘッダ７７０'のコード（ＣＯＤＥ）部７４０のＴＲＣマッピングフィールドを、例えば、図１１に示されるヘッダ７７０'のコード（ＣＯＤＥ）部７４０'のＴＲＣマッピングフィールドに、変更することにより実行される。このようにして、復号器は、新しい画素値に基づいてブロックをマッピングする。

【００５０】図１３は、ビットデプスを４ビット／画素から２ビット／画素に変更するためのビットデプス変換操作を示す。ビットデプス、すなわち１画素当りのビットの数、は単にヘッダのビットデプスフィールドを変更することにより、変更される。従って、出力画像を処理するために、各出力画素において、指示された数の最上位のグレースケール値ビットのみが保持される。この操作は、所与の数のビットのビットデプスを用いるが、異なった数のビットのビットデプスを有する装置を用いてディスプレイ又はプリントされる、スキャナ及びカメラなどの装置に有用である。このような操作において、復号化コマンドは、命令においてビットデプスを変更するために用いられる。図１１の、ヘッダ７７０'において、復号化コマンドがビットデプス値を「８」から「２」に変更するために用いられる。

【００５１】図１４は、リサイジング（再サイズ調整）操作を示す。この操作は、ブロックの大きさを拡大又は縮小するために、空間的スケーリングによって行われる。もしも可能であれば、復号器５００は、空間的に変更されるサイズのブロックを可能性として用いることにより、要求されたサイズに近づくよう試みるであろう。本発明の実施形態に従って、画像データよりもむしろ、コードブックエントリがリサイズされる。これは、復号化コマンドを用い、且つ、ビット高値及びビット幅値に整数でない数を用いて変更することにより行われる。例えば、図１４の部分（ａ）に示されるブロック１１～１９が３分の２にリサイズされる場合、復号器は、図１４の部分（ｂ）に示されるブロック１１'～１９'に収まる、復号化された画像ブロックを出力するために、コードブックエントリを均等にスケーリングすることにより、それ（ブロック１１～１９）を必要な大きさに近づけることを試みる。復号器がブロックを均等に必要な大きさに近づけることができない場合、復号器は、３分の２のリサイジングを得るために、コードブックエントリを部分（ｃ）に示されるようなブロック１１"～１９"にスケーリングすることを試みる。図１１は、復号化コマンドがビット高値とビット幅値を「４」から「２」に変更するために用いられるヘッダを示す。これは、画像のサイズを、各方向に半分に、すなわち元の領域を４分の１に、効果的にカットする。

【００５２】図１５は、画素１１～１９を有する元の画像の９０°の回転を示す。回転後、図１５の部分（ａ）に示される元の画像は、図１５の部分（ｂ）に示される、画素１１'～１９'を有する画像に変換される。この操作は、あらゆる高速なアルゴリズムを用いてコードワードを回転させることによって行われる。復号化コマンドは、ブロックの画定された再配置順序を変更し、ブロック内又はブロック相互間の回転を反映するように用いられる。この操作は、変換された画素２１～２９及び３１～３９をそれぞれに形成するために、図１６及び図１７に示されるミラリング（鏡映）及び転置操作にも実行される。図１１のヘッダ７７０'は、図１０に示されるヘッダ７７０'と比較して、走査フィールドを変更することにより、画素の順序を変更する。再構成の際、復号器５００は、コード部７４０の走査フィールド上に定められる走査順序に従って、コードブックのブロックを実際の出力画像ブロックに写像する。

【００５３】図１に示されるように、符号器４００は、プログラムされた汎用コンピュータで実施されるのが好ましい。しかしながら、符号器４００は、専用コンピュータ；プログラムされたマイクロプロセッサ又はマイクロコントローラ及び周辺集積回路エレメント；ＡＳＩＣ又は他の集積回路；デジタルシグナルプロセッサ；ディスクリート（個別）エレメント回路などのハードワイヤード電子又は論理回路；ＰＬＤ、ＰＬＡ、ＦＰＧＡ、又はＰＡＬなどのプログラム可能論理装置；などで実施されることができる。一般に、有限状態のマシンを実行することが可能な、つまり、図９に示されるフローチャートのステップＳ１１００～Ｓ１３００を実行することが可能な、あらゆるデバイスが符号器４００を実施するために用いられることができる。

【００５４】図１に示されるように、復号器５００は、プログラムされた汎用コンピュータで実施されるのが好ましい。しかしながら、復号器５００は、専用コンピュータ；プログラムされたマイクロプロセッサ又はマイクロコントローラ及び周辺集積回路エレメント；ＡＳＩＣ又は他の集積回路；デジタルシグナルプロセッサ；ディ

15

スクリート（個別）エレメント回路などのハードワイヤード電子又は論理回路；ＰＬＤ、ＰＬＡ、ＦＰＧＡ、又はＰＡＬなどのプログラム可能論理装置；などにも与えられることができる。一般に、有限状態のマシン（装置）を実施することが可能な、つまり、図９に示されるフローチャートのステップＳ１３００～Ｓ１５００及び／又は図１２～１７に示される操作を実行することが可能な、あらゆるデバイスが、復号器５００を実施するために用いられることができる。

【００５５】１つの好ましい実施形態は、復号化コマンドを含む追加のヘッダを有する圧縮された画像データファイルを圧縮するが、圧縮された画像データファイルに復号化コマンドを採り入れるための知られている又は後に発明される方法の何れもが本発明の範囲に含まれる。つまり、本発明は、本明細書に開示される復号化コマンドを採り入れるための特定の方法に限定されず、圧縮された画像データファイルに復号化されたコマンドを採り入れる方法にかかわらず、更に概括的に復号化コマンドを有する圧縮された画像データファイルを含む。

【図面の簡単な説明】

【図１】本発明に従った符号化及び復号化システムの汎用化されたブロック図である。

【図２】従来のＶＱ符号器を示す。

【図３】従来のＶＱ復号器を示す。

【図４】８つの画素が最終的なコードワードに量子化される工程を含むＨＶＱの実施例を示す。

【図５】２つの入力記号をコードワードに符号化するＨＶＱ方法を示す。

16

＊【図６】受け取られたコードワードに基づいて入力記号を再構成するＨＶＱ方法を示す。

【図７】本発明に従ったＶＱ符号器を示す。

【図８】本発明に従ったＶＱ復号器を示す。

【図９】本発明に従った画像の符号化、処理及び復号化の方法の概要を記述するフローチャートである。

【図１０】本発明に従ったコードブック操作方法のヘッダを示す。

【図１１】本発明に従って、追加された復号化コマンドを用いたコードブック操作方法のヘッダを示す。

【図１２】本発明に従って、追加された復号化コマンドを用いて実行される処理操作の実施例を例示する。

【図１３】本発明に従って、追加された復号化コマンドを用いて実行される処理操作の実施例を例示する。

【図１４】本発明に従って、追加された復号化コマンドを用いて実行される処理操作の実施例を例示する。

【図１５】本発明に従って、追加された復号化コマンドを用いて実行される処理操作の実施例を例示する。

【図１６】本発明に従って、追加された復号化コマンドを用いて実行される処理操作の実施例を例示する。

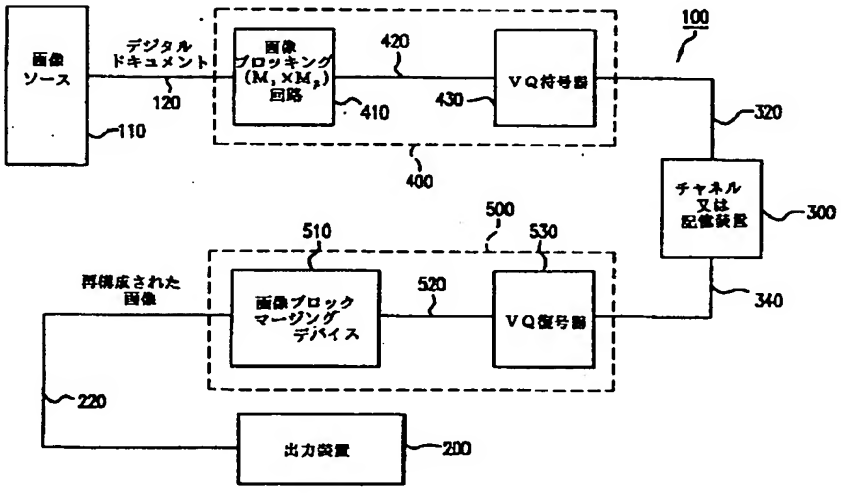
【図１７】本発明に従って、追加された復号化コマンドを用いて実行される処理操作の実施例を例示する。

【符号の説明】

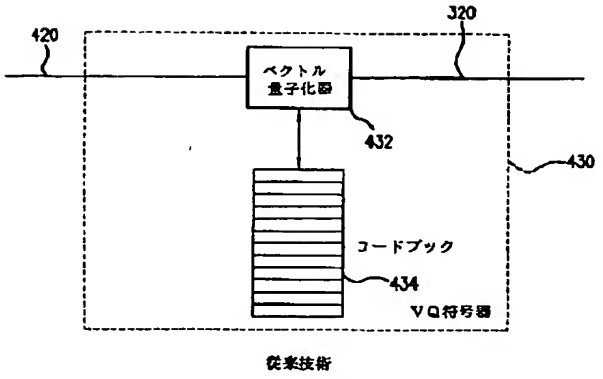
１１０　画像ソース
４２０　信号ライン
４３０　ＶＱ符号器
５３０　ＶＱ復号器

＊

【図１】

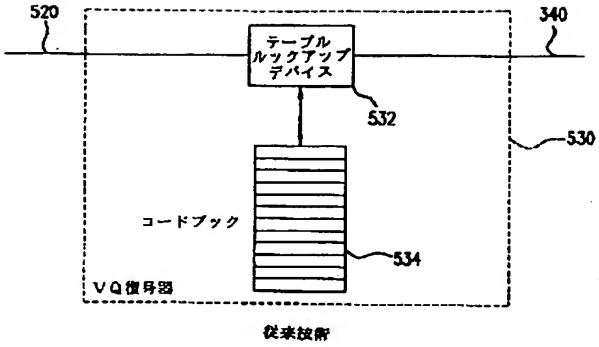


【図２】



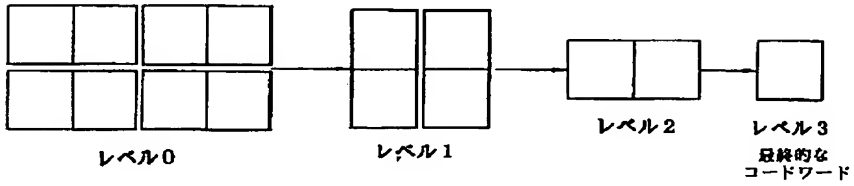
従来技術

【図３】

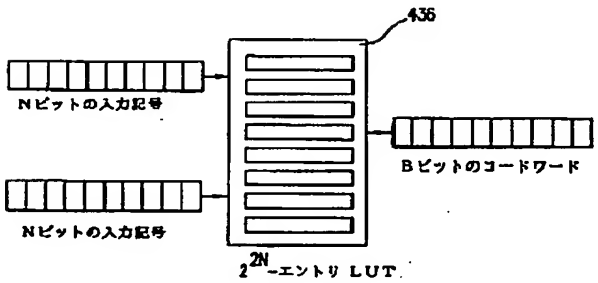


従来技術

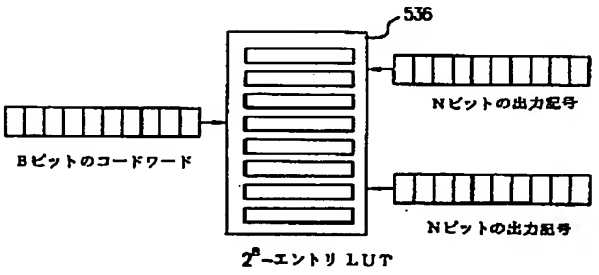
【図４】



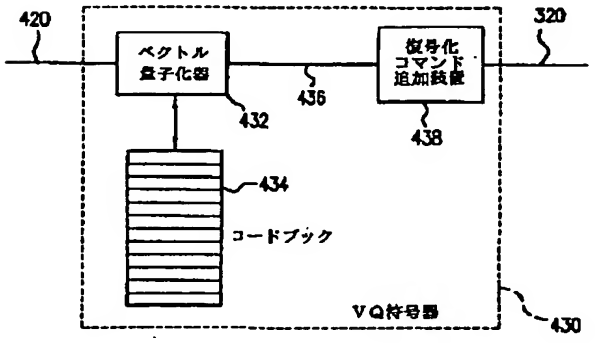
【図５】



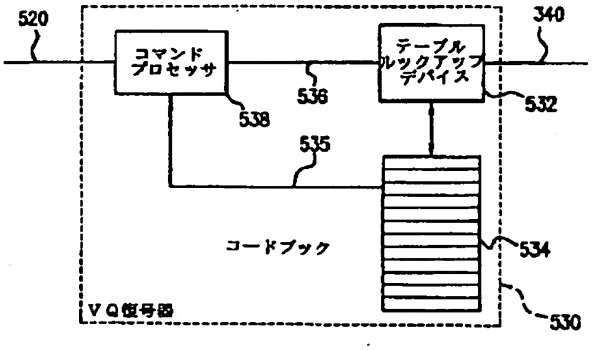
【図６】



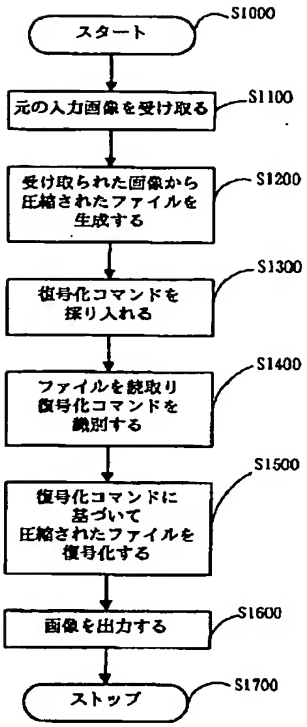
【図７】



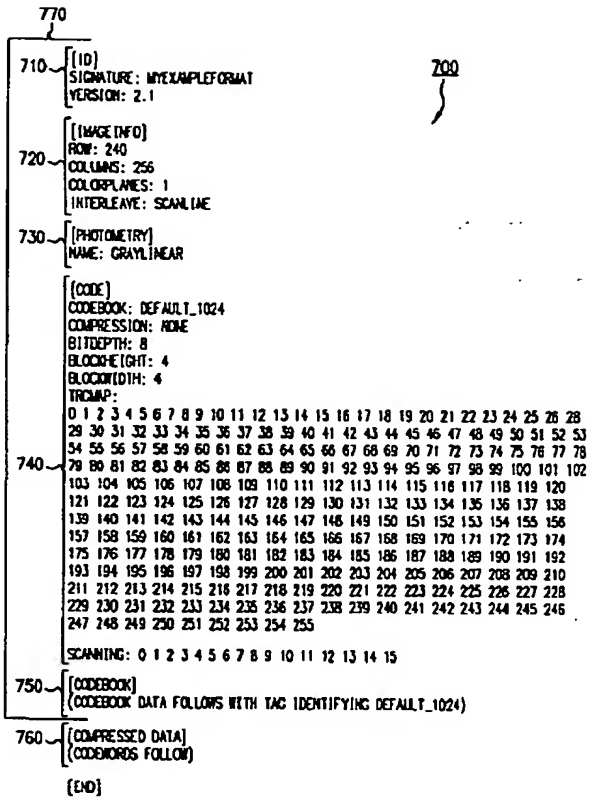
【図８】



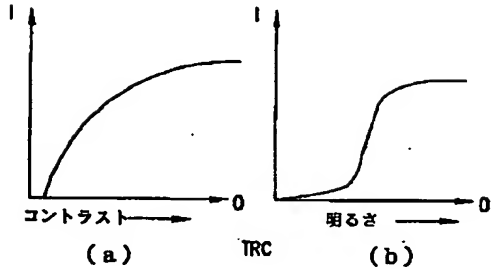
【図９】



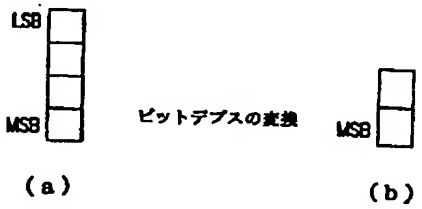
【図１０】



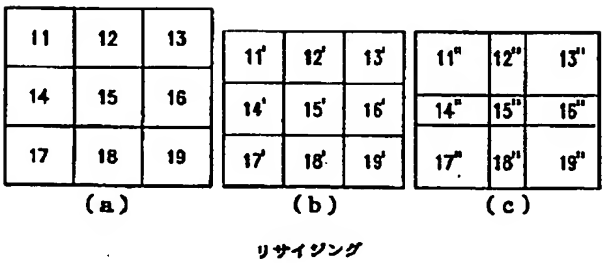
【図１２】



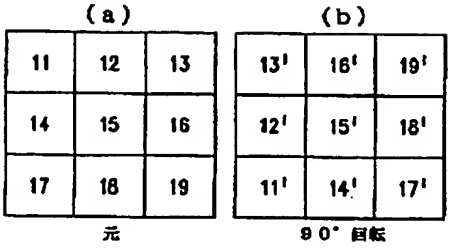
【図１３】



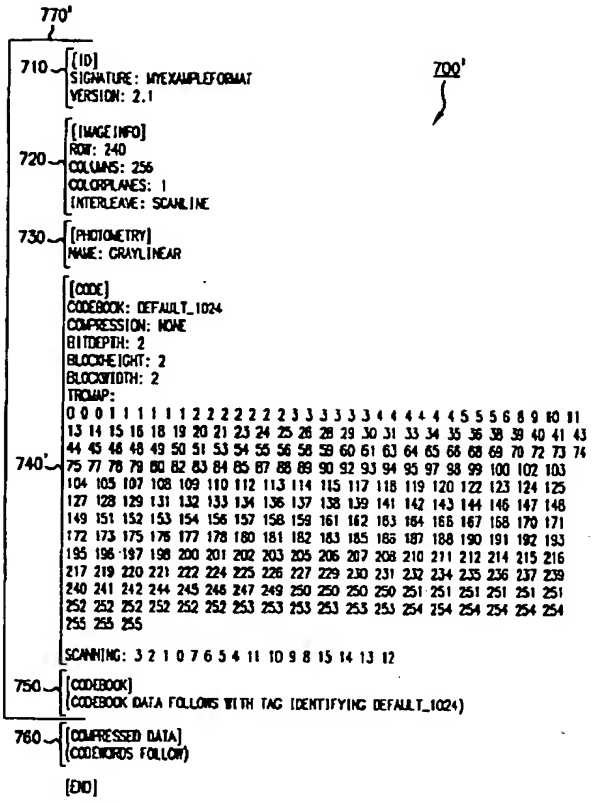
【図１４】



【図１５】



【図１１】



【図１６】

(a)

21	22	23
24	25	26
27	28	29

元

(b)

27'	28'	29'
24'	25'	26'
21'	22'	23'

垂直方向の鏡映（ミラー）

【図１７】

(a)

31	32	33
34	35	36
37	38	39

元

(b)

31'	34'	37'
32'	35'	38'
33'	36'	39'

転置

【外国語明細書】

A COMPRESSION FRAMEWORK INCORPORATING DECODING COMMANDS

This invention relates generally to processing compressed digital images. More particularly, this invention is directed to methods and apparatus for incorporating decoding commands into a compressed image data file to complement processing performed in the compressed domain to reduce computation time for imaging operations.

Data compression is required in data handling processes, where too much data is present for practical applications using the data. Commonly, compression is used in communication links to reduce the transmission time or required bandwidth. Similarly, compression is preferred in image storage systems, including digital printers and copiers, where "pages" of a document to be printed are stored temporarily in precollation memory. The amount of media space on which the image data is stored can be substantially reduced with compression. Generally speaking, scanned images, i.e., electronic representations of printed documents, are often large, and thus make desirable candidates for compression.

This invention is directed to reducing computation time for imaging operations that alter the appearance of the original image. In particular, this invention introduces decoding commands into a compressed image data file. These commands are instructions carried along to the decompressor and complement processing performed in the compressed domain. The commands, possibly along with original image parameters, instruct the decompressor to alter the appearance of the decoded image relative to the original image. Then, the decompressor performs some processing steps to the data while decompressing it. This results in reduced computation time for the image operations.

One aspect of this invention deals with a basic problem in the frequent use of compressed image buffers – the inability to process image data without performing expensive decompression and recompression operations. To make processing in the compressed domain more accessible, the compression format/methodology according to this invention preferably generates compressed data in such a way that a predetermined amount of data corresponds to a predetermined number of pixels in the image. If this condition holds, image regions in the compressed data can be easily

identified, replaced or cropped. One preferred embodiment of this invention includes a method for compressing an image and a syntax for decoding commands that allows some image processing operations to be easily applied in the compressed domain.

A preferred embodiment of the system and method of this invention works particularly well for compression schemes such as vector quantization (VQ). In vector quantization, a group of pixels is quantized at the same time by representing the group as a code vector. The image data are first processed into sets of vectors. A codebook that best matches the data to be quantized is then generated. Each input vector is then quantized to the closest codeword. Compression is achieved by transmitting only the indices for the codewords. At the receiver, the images are reconstructed using a table look-up procedure.

In a preferred embodiment of the system and method of this invention, both the coder and the decoder share codebooks. In this embodiment, the codewords take the form of auxiliary data, and the compressed image data is generated based on the auxiliary data. The encoding indices of the auxiliary data are transmitted, and from these encoding indices, the decoder looks up a set of pixels. Thus, to process the compressed image, a decoding command can be incorporated into the compressed image data file. The decompressor that receives the decoding commands understands the commands and executes them over the codebook entries and/or the compressed image data. The auxiliary data is processed based on the decoding commands. The decoding of the compressed image data is based on the processed auxiliary data. The image is decompressed normally because the codebook entries are irrelevant for the decoding process, as long as the correspondence between the indices and the codebook entries is maintained.

Accordingly, the method and system of this invention can apply any number of operations to the compressed data, including TRC mapping, bit-depth conversion, resizing, rotation, mirroring, transposition, halftoning, cropping, pasting and merging. The system of this invention can include a data or image processing system able to compress and/or decompress an image. The method and system described herein are advantageous because they result in reduced computation time for imaging operations.

These and other features and advantages of this invention are described in or are apparent from the following detailed description of the preferred embodiments.

The preferred embodiments of this invention will be described in detail, with reference to the following figures, wherein:

Fig. 1 is a generalized block diagram of an encoding and decoding system according to this invention;

Fig. 2 shows a conventional VQ coder;

Fig. 3 shows a conventional VQ decoder;

Fig. 4 shows an example of HVQ involving eight image pixels quantized into the final codeword;

Fig. 5 shows an HVQ method of encoding two input symbols into a codeword;

Fig. 6 shows an HVQ method of reconstructing input symbols based on a received codeword;

Fig. 7 shows a VQ coder according to this invention;

Fig. 8 shows a VQ decoder according to this invention;

Fig. 9 is a flowchart outlining an image encoding, processing and decoding method according to this invention;

Fig. 10 shows a header in a codebook operation method according to this invention.

Fig. 11 shows a header in a codebook operation method using appended decoding commands according to this invention.

Figs. 12-17 illustrate examples of the processing operations that can be performed using appended decoding commands according to this invention.

Figure 1 shows a generalized functional block diagram of an encoding and decoding system 100 according to this invention. The system 100 includes an image source 110 that may be any one of a number of different image sources, such as a scanner, a digital copier or a facsimile machine device, that are suitable for generating electronic image data, or a device suitable for storing and/or transmitting electronic image data, such as a client or server of a network. The electronic image data output from the image source 110 is provided to the encoder 400 via a signal line 120. The encoder 400, while preferably providing additional functionality, is capable of carrying out image processing and compression operations in accordance with this invention. It should be appreciated that, while Fig. 1 shows the encoder 400 as a VQ encoder, according to this invention any type of lossless or lossy image compression

can be performed by the encoder 400. In such a case, the particular elements of the encoder 400 may be different than those shown in Fig. 1. Thus, this invention is not limited to VQ encoding, and specifically includes any known or later developed type of image encoding or compression.

In one preferred embodiment shown in Fig. 1, the encoder 400 includes an image blocking portion 410 that divides electronic image data into a plurality of $M_1 x M_2$ blocks or segments. In one embodiment, the blocking operation may be accomplished by a windowing or cropping circuit that enables the transfer of data comprising one or more $M_1 x M_2$ blocks of data from the input document to an $M_1 x M_2$ data buffer in the encoder 400. Once stored in the data buffer, the $M_1 x M_2$ blocks of image data are next sent via a signal line 420 to be operated on by a VQ coder 430. In the VQ coder 430, the data may be operated on using various compression operations associated with the vector quantization technique to compress the image data within a block.

Once compressed, each $M_1 x M_2$ block of image data is then preferably transmitted to a channel or storage device 300 via a signal line 320. The channel or storage device 300 can be either or both of a channel device for transmitting the compressed image data to the decoder 500 or a storage device for indefinitely storing the compressed image data until there arises a need to decompress the compressed image data. The channel device can be any known structure or apparatus for transmitting the compressed image data from a first apparatus implementing the encoder 400 according to this invention to a physically remote decoder 500 according to this invention via a signal line 340. Thus, the channel device can be a public switched telephone network, a local or wide area network, an intranet, the Internet, a wireless transmission channel, or the like or any other distributed network.

Similarly, the storage device can be any known structure or apparatus for indefinitely storing compressed image data, such as RAM, a hard drive and disk, a floppy drive and disk, flash memory or the like. Moreover, the storage device can be physically remote from the encoder 400 and/or the decoder 500, and reachable over the channel device described above.

The output of the channel or storage device 300 is a coded or compressed unit of data to be decoded by the decoder 500. It should be appreciated that, similarly to

the encoder 400 discussed above, while Fig. 1 shows the decoder 500 as a VQ decoder, according to this invention any type of lossless or lossy decompression can be performed by the decoder 500, so long as it matches the type of compression used to generate the compressed image data file to be decoded. In such a case, the particular elements of the decoder 500 may be different than those shown in Fig. 1. Thus, this invention is not limited to VQ decoding, and specifically includes any known or later developed type of image decoding or decompression.

In the preferred embodiment shown in Fig. 1, the decoder 500 includes a VQ decoder 530 that receives the compressed M_1xM_2 block of data from the channel or storage device 300 via the signal line 340 and an image processor 510 that receives the decoded image from the VQ decoder 530 via a signal line 520. Though the decoder 500 is shown in Fig. 1 as physically separate from the encoder 400, it should be understood that the decoder 500 and the encoder 400 may be different aspects of a single physical device.

The reconstructed image output from the image processor 510 of the decoder 500 can be transmitted to an output device 200 via a signal line 220. It should be understood that the output device 200 can be any device that is capable of outputting or storing an image processed in accordance with this invention.

Fig. 2 shows conventional VQ coder. In the conventional VQ coder 430, an M_1xM_2 block of image data is input to the vector quantizer 432. In the vector quantizer 432, the input block of image data is compared to the entries stored in an encoding codebook 434. The entry in the codebook 434 closest to the input block is selected. The quantizer vector 432 thus outputs only the index of the codebook for the selected codeword entry for the input block of image data. This index, or codeword, can optionally be further encoded using entropy coding, such as, for example, arithmetic encoding, Huffman encoding, LZW encoding, and the like.

Fig. 3 shows a conventional VQ decoder 530. In the conventional VQ decoder 530, the B-bit codewords are input to a table look-up device 532, where the index is input to a decoding codebook 534. The pixel pattern entry in the codebook 534 corresponding to the index is output from the table look-up device 532 as the decoded image block.

An example of a practical VQ implementation is hierarchical VQ (HVQ). Fig. 4 shows an example of a HVQ implementation involving eight image pixels of a zero-th level (LEVEL 0) of an input image quantized into the final codeword. The eight image pixels of the zero-th level of the input image are passed through one state of vector quantization to form four first-level codewords (LEVEL 1) of an input image. The four codewords of the first-level of an input image are quantized into two new second-level codewords (LEVEL 2) of an input image. The two new codewords of the second level of an input image are, finally, quantized into a third-level codeword (LEVEL 3) of an input image. In this example, seven table look-ups are used. While the HVQ coder 430 needs to perform successive table look-ups, the HVQ decoder 530 does not need to follow the opposite path, since the final codeword fully specifies the $M_1 x M_2$ block of the decoded image.

Fig. 5 shows an example of HVQ implementation in which, at each step, two symbols of N bits are encoded using a fixed B-bit codeword through a look-up table (LUT) 436. After the codeword is designated, the codeword is transmitted. As shown in Fig. 6, a decoder is able to reconstruct the two N-bit symbols based on the received codeword through a look-up table (LUT) 536.

In HVQ implementation, at each step, two symbols are combined into an output encoded symbol, reducing the number of symbols by a factor of two. At each step, the number of symbols is reduced. Thus, after S steps, the number of symbols is reduced, by a factor of 2^S, to a single symbol. Compression is achieved since a reduced number of bits is actually transmitted. With a good design technique, the goal is to minimize the distortion incurred by the process. The advantage is the computational simplicity, since only stages of tables for conversion from 2N to 2B bits need to be provided.

Compression schemes such as hierarchical vector quantization are very fast for both encoding and decoding data. They are also very practical for the use of decoding commands, as embodied in an embodiment of this invention.

It should be understood that, in general, a conventional compressed image data file includes the compressed image data and additional information that provides the original image parameters. The decoder requires these original image parameters to structure the decompressed image data. That is, without this additional information,

the decoder would be unable to determine how to structure the decoded image data to form an image. This additional information generally includes one or more of the image information and photometry sections 720 and 730 shown in Figs. 10 and 11. This additional information is generally provided in a header 770 that is combined with the compressed image data 760 to form the compressed image data file 700, as shown in Figs. 10 and 11.

It should be appreciated that the decoding commands according to this invention include any information such that, when the decoding commands are added to the compressed image data file, the decoding commands allow, in combination with the conventional additional information provided in the header to the compressed image data file, the output image formed from decompressing the compressed image data to be different from the original image from which the compressed image data was created. It should be further appreciated that the output image can differ from the original image in any way, including, but not limited to, orientation, size, scale, aspect ratio, bit depth, and any combination of these and other alterations that could be made to the original image.

It should even further be appreciated that the decoding commands can be added to the compressed image data file at any time between the time the compressed image data file is created by the encoder 400 and the compressed image data file is decoded by the decoder 500.

With the decoding commands according to this invention added to the header, the compressed data file specifies how the decompressed image data is to be interpreted and/or what functions are to be performed by or on the codebook entries of the decoding codebook 534 by the decoder 500. As the codebook 534 is often small compared to the image, operations on the codebook 534 are generally computationally inexpensive. Similarly, changing how the decompressed data is to be interpreted is also generally computationally inexpensive. Once the decoder 500 reads the input data file and identifies the decoding commands, the decoder 500 changes the fields of the input data file and decodes the compressed image data in accordance with the decoding commands by, for example, operating over the codebook. For every codebook entry, a block of $M_1 x M_2$ pixels is processed.

As shown in Fig. 7, in accordance with one embodiment of this invention, in the encoder 400, the codewords output by the vector quantizer 432 as auxiliary data are input to a decoding command appender 438 via line 436, where decoding commands are introduced and appended to, or more generally, incorporated into, the additional information added to the compressed image data to form the header of the compressed image data file.

In one preferred embodiment of this invention, the decoding commands appender adds or makes changes to a CODE portion 740 of the header 770, as shown in Figs. 10 and 11. The code portion contains one or more data fields, such as a bit-depth field, a bit-width field, and the like.

The data for each of these fields is originally set based on the original image data and/or the original encoding scheme and codebook used to generate the indices. Thus, for example, for 8-bit byte map or continuous tone image data, the bit-depth field of the header is originally set to "8". Similarly, if the image data is encoded using a block width of 4 and a block height of 4, the block width and block height fields of the header are originally set to "4". Likewise, if a particular codebook is used to generated the indices, the codebook field is set to a label that either refers to a codebook that is explicitly set forth in the header or, if the codebook is already known to the decoder, identifies the codebook to be used to decode the compressed image data.

Then, by changing the values of one or more of the data fields in the CODE portion 740 of the header from the original values determined from the image data and/or the encoding scheme or codebook, the decoder, when decoding the compressed image data based on the changed values of the data fields in the CODE portion 740 of the header, modifies the codewords, and decodes the compressed image data based on the modified codewords to form the modified decompressed image.

As shown in Fig. 8, in accordance with this invention, in the decoder 500, the codebook indices and appended decoding commands are sent to the command processor 538, where the codewords are read and the decoding commands are identified. The fields changed in the header can include, for example, the bit-depth (BD), i.e., the desired amount of bits per pixel in the reconstructed image, the block-width (BW), i.e., the width in pixels of the reconstructed block, the block-height

(BH), i.e., the height in pixels of the reconstructed block, and the like. The codewords are then processed based on the decoding commands. The output from the command processor 538 is sent to the table look-up device 532 over a signal line 536. Then the table look-up device 532 proceeds to operate over the codebook 534. For example, new codebook data, either explicitly provided in the header, or identified by name in the header, is provided from the command processor 538 over a signal line 535 to the codebook 534, replacing the previously stored codebook data. The compressed image data is accordingly decoded based on the processed codewords.

It should be appreciated that, as set forth above, the decoding command appender 438 could be included in the VQ decoder 530, instead of or in addition to being provided in the VQ encoder 430. In this case, when the decoding command appender 438 is provided in the decoder 500, it inputs the received compressed image data file over the signal line 520. This received compressed image data file may or may not already have decoding commands appended to it. The decoding command appender 438 in the VQ decoder 530 inputs the received compressed image data file and adds decoding commands to, or changes the decoding commands already in the header of, the compressed image data file to obtain a desired output image that differs from the original image in a desired way. The revised compressed image data file is then output by the decoding commands appender 438 to the command processor 538, as described above. Similarly, the decoding commands could alternatively be appended at any time between being output by the encoder 400 and input by the decoder 500.

A decoding command set preferably includes instructions for scaling the block to fit a particular size, rearranging the block entries, and performing pixel-wise operations. Scaling the block to fit a particular size includes, for example, the operations of altering the bit-depth, altering the block-width, which was originally M_1 pixels, or altering the block-height, which was originally M_2 pixels. Rearranging the block entries can be accomplished by using a vector v with mn numbers. If the original blocks are arranged in the natural scan order, i.e., left-to-right or top-to-bottom into a vector x, rearranging the block entries results in the vector y. The mapping from x to y is controlled by:

$y[k]=x[v[k]]$ for $k=1,\ldots,mn$.

It should be appreciated that more complex commands, such as program code whose input is a block of entries for the decoding codebook, can be included in the appended decoded commands. For example, the commands can be given in JAVA™ code, or any other predetermined syntax.

Fig. 9 is a flowchart outlining an image encoding, processing and developing method according to this invention. Beginning at step S1000, control continues to step S1100, where electronic image data for an original image is input. Then, in step S1200, compressed image data is generated from the electronic image data. Next, in step S1300, the decoding commands are introduced to the compressed image data. Control then continues to step S1400.

In step S1400, the data is read and the decoding commands are identified. Next, in step S1500, upon identification of the decoding commands, the decoder proceeds to decode the compressed image data based on the identified decoding commands. This step can include, for example, operating over the codebook. Then, in step S1600, a resulting image is output. Then in step S1700, the process stops.

Figs. 10 and 11 show in greater detail how the codebook is operated on in step S1500. Fig. 10 shows an exemplary compressed image data file 700, including an exemplary header 770 and the compressed image data 760. The compressed image data file 700 was generated from a monochrome 960 pixels x 1024 pixels original image, which was compressed using VQ encoding with blocks of 4 pixels x 4 pixels so that each 4 pixels x 4 pixels block is mapped to a codeword. In particular, while Fig. 10 illustrates the various decoding commands, these decoding commands do not result in any alterations to the decoded image compared to the original image.

The ID portion 710 of the header 700 identifies the format to the decoder 500. The IMAGEINFO portion 720 of the header 700, as discussed above, provides the original image parameters, as the additional information to the decoder 500. In particular, as shown in Fig. 10, the IMAGEINFO portion 710 indicates that original image is a 240x256 array of pixels and that there is only one color plane, i.e., that the image is monochrome. The PHOTOMETRY portion 730 of the header 700, as discussed above, provides the original photometry, or color space, to the decoder 500

to be used when decoding the compressed image data. As shown in Fig. 10, the original photometry, or color space, to be used by the decoder 500 is gray-linear photometry.

The CODE portion 740 of the header 700 provides the decoding commands that allows output image to be altered relative to the original image. As discussed above, the CODE portion 740 includes fields such as the output codebook, the compression, the output bit-depth, the output block-height, the output block-width, the output TRC-mapping, and the output scanning order. By changing the values of these fields from the values used when compressing the image, the decoder 500 processes the sets of output pixels corresponding to the transmitted codewords or codeword indices to convert the output pixels to have altered values based on the altered ones of these fields. This will be explained in greater detail below with respect to Fig. 11.

In particular, as shown in Fig. 10, the CODE portion 740 indicates that the output image is to be decoded as blocks of 4 pixels x 4 pixels with 8 bits per pixel, passing through a linear map in the natural scanning order. The codebook is set to a particular codebook label, "Default_1024". As described above, this label can refer either to a codebook explicitly defined in the codebook portion 750 of the header 770, or to a codebook that is known to be available to a generic decoder 500 or the specific decoder 500 that is going to be used to decode the compressed data 760. If the particular codebook as set is not present, a predetermined codebook is used. There is no compression of the codeword array.

The codebook portion 750 of the header 770, if not empty, sets forth the actual codebook entries, as shown in Fig. 10. The compressed data portion 760 of the compressed image data file 700 sets forth the actual codebook entry indices for the 61440 (240x256) blocks of encoded image data.

It should be appreciated that the codebook label provided in the CODE portion 740' can refer to a codebook that is different from the codebook used to compress the original image data. Moreover, while the codebook used to encode the original image data can use the same label as the codebook identified by the codebook label of the code portion 740, the two codebooks do not have to be the same. Thus, the codebook portion 750, while nominally having the same label as the codebook used to compress or encode the original image data, can explicitly define a codebook that is different

from the codebook used to compress or encode the original image data. Similarly, even if the codebook label of the code portion 740 merely identifies a codebook already available to the decoder 500, that identified codebook can nonetheless be different from the codebook used to compress or encode the original image data.

Fig. 11 is an exemplary compressed image data file 700' that includes a header 770' that changes the appearance of the output image upon decoding the compressed data file . By changing some of the header fields in the CODE portion 740', the image data blocks are processed accordingly by the decoder 500 to provide an altered decoded image. In Fig. 11, the bit-depth field is changed from "8", as shown in Fig. 10, to "2", the bit-height field is changed from "4", as shown in Fig. 10, to "2", and, the bit-width field is changed from "4", as shown in Fig. 10, to "2". Thus, compared to the values for these fields used when generating the codewords in the VQ coder 400, these values result in an output image that has one-half (2/4) the width, or horizontal extent, of the input image, one half (2/4) the height, or vertical extent, of the input image, and one-quarter (2/8) the bit-depth, or greyscale levels, of the input image.

In Fig. 11, the TRC-mapping field and the scanning order field are also altered compared to Fig. 10. In particular, in the CODE portion 740' shown in Fig. 11, the TRC field alterations change the greyscale values of the decoded pixels from the initial greyscale values shown in Fig. 10. In Fig. 10, each greyscale value of the output image corresponds to the position, starting from "0", of that value in the list shown in the TRC field. In Fig. 11, the positions below 128 have values moved towards zero from their positional value. In contrast, the positions above 127 have their values moved towards 256 from their positional values. The effect of these value shifts is to increase the contrast between the lighter and darker pixels.

In addition, in the CODE portion 740' shown in Fig. 11, the scanning field alterations change the placement of the blocks according to their position in the scan line from the input order, represented by the scanning field shown in Fig. 10, to the output order, defined by the scanning field shown in Fig. 11.

It should be appreciated that other fields may be changed, and that although all of the bit-depth, bit-height, bit-width, TRC-mapping and scanning fields are shown as being changed in Fig. 11, each can be changed independently.

Additional decoding commands can also be added to the headers 770 and 770' shown in Figs. 10 and 11. These additional decoding commands can be in addition to, or in place of, the decoding commands of the CODE portion 740 and/or the changes made to the codebook portion 750.

These additional decoding commands are used to change the original image parameters. These changes to the original image parameters are implemented by adding decoded image parameters to the IMAGEINFO portion 720 and/or the Photometry portion 730, rather than changing the image parameters initially set forth in the IMAGEINFO and Photometry portions 720 and 730. That is, to use these additional decoding commands, the command processor 538 must have both the original image parameters and desired image parameter decoding commands for the desired output image. The command processor 538 uses the original and desired image parameter decoding commands, to determine how the compressed image data in the compressed data portion 760 is to be decompressed so that the decompressed image data has the desired image parameters.

Figs. 12-17 show examples of processing operations that can be performed using the appended decoding commands according to this invention. Fig. 12 shows a tone-reproduction curve (TRC) as a pixel-wise operation. In this operation, every input pixel value is mapped to another pixel value on the curve. In portion (a) of Fig. 12, the contrast of the output image is decreased relative to the input image. In portion (b) of Fig. 12, the brightness of the output image is increased. To enhance contrast or brightness of an image using the example header of Figs. 10-11, TRC pixel-wise operation is performed by changing the TRC-mapping field in the CODE portion 740 of the header 770' shown in Fig. 10, for example, to the TRC-mapping field in the CODE portion 740' of the header 770' shown in Fig. 11. Accordingly, the decoder maps the blocks based on the new pixel values.

Fig. 13 shows a bit-depth conversion operation to change the bit-depth from 4 bits/pixel to 2 bits/pixel. The bit-depth, i.e., the number of bits per pixel, can be changed by simply changing the bit-depth field of the header. Accordingly, to process the output image, in each output pixel, only the indicated number of most-significant greyscale value bits are retained. This operation is useful for devices such as scanners and cameras, which use a bit-depth of a given number of bits, but will be displayed or

printed using a device having a bit-depth of a different number of bits. In such operations, the decoding command is used to change the bit-depth in the instructions. In Fig. 11, in the header 770', the decoding command is used to change bit-depth value from "8" to "2".

Fig. 14 shows a resizing operation. This operation is done by spatial scaling to enlarge or reduce the size of the blocks. The decoder 500 will try to approximate the requested size by possibly using blocks of spatially varying size if possible. According to an embodiment of the invention, the codebook entries, rather than the image data, are resized. This is done by modifying using decoding commands and using non-integer numbers for the bit-height and bit-width values. For example, if blocks 11-19 shown in portion (a) of Fig. 14 are to be resized by two-thirds, the decoder will try to approximate it by scaling the codebook entries uniformly to output decoded image block that fit into blocks 11'-19', as shown in portion (b) of Fig. 14. If the decoder cannot approximate the blocks uniformly, it will try to scale the codebook entries into blocks 11"-19" as shown in portion (c) to obtain a resizing of two-thirds. Fig. 11 shows a header in which the decoding command is used to change bit-height and bit-width values from "4" to "2". This effectively cuts the size of the image in half in each direction, or one-quarter the original area.

Fig. 15 shows a rotation of original image by 90° having pixels 11-19. After the rotation, the original image shown in portion (a) of Fig. 15 is transformed into the image having pixels 11'-19', shown in portion (b) of Fig. 15. This operation is done by rotating the codewords using any fast algorithm. The decoding command is used to modify a defined rearrangement order of the blocks, to reflect an intrablock or interblock rotation. This operation is also performed on the mirroring and transposition operations shown in Fig. 16 and Fig. 17 to form transformed pixels 21-29 and 31-39, respectively. The header 770' of Fig. 11 changes the order of the pixels by changing the scanning field, compared to the header 770' shown in Fig. 10. At reconstruction, the decoder 500 maps the codebook block to the actual output image block according to the scanning order defined on the scanning field of the code portion 740.

As shown in Fig. 1, the encoder 400 is preferably implemented on a programmed general purpose computer. However, the encoder 400 can also be

implemented on a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit elements, an ASIC or other integrated circuit, a digital signal processor, a hardwired electronic or logic circuit such as a discrete element circuit, a programmable logic device such as PLD, PLA, FPGA or PAL, or the like. In general, any device, capable of implementing a finite state machine that is in turn capable of implementing steps S1100-S1300 of the flowchart shown in Fig. 9, can be used to implement the encoder 400.

As shown in Fig. 1, the decoder 500 is preferably implemented on a programmed general purpose computer. However, the decoder 500 can also be implemented on a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit elements, an ASIC or other integrated circuit, a digital signal processor, a hardwired electronic or logic circuit such as a discrete element circuit, a programmable logic device such as a PLD, PLA, FPGA or PAL, or the like. In general, any device, capable of implementing a finite state machine that is in turn capable of implementing steps S1300-S1500 of the flowchart shown in Fig. 9 and/or the operations shown in Figs. 12-17, can be used to implement the decoder 500.

While one preferred embodiment compresses a compressed image data file having an appended header containing the decoding commands, any known or later developed method for incorporating the decoding commands into the compressed image data file is within the scope of this invention. That is, this invention is not limited to the specific methods disclosed herein for incorporating the decoding commands into the compressed image data file, but more generally encompasses a compressed image data file that includes decoding commands, regardless of the method for incorporating the decoding commands into the compressed image data file.

WHAT IS CLAIMED IS:

1. A method for processing compressed image data, comprising:

receiving electronic image data representing an original image;

generating a compressed image data file from the received electronic image data;

incorporating decoding commands into the compressed image data file, the decoding commands comprising instructions which alter an appearance of a decoded image relative to the original image;

identifying the decoding commands; and

decoding compressed image data contained in the compressed image data file according to the identified decoding commands to form processed electronic image data of the decoded image, the decoded image representing an altered version of the original image processed.

2. A system for processing compressed image data, comprising:

an image source that outputs electronic image data representing an original image;

a coder that inputs the electronic image data, comprising:

a compressor that generates a compressed image data file from the electronic image data, and

an appender that incorporates decoding commands into the compressed image data file, the decoding commands comprising instructions which alter an appearance of a decoded image relative to the original image;

a decoder that decodes the compressed image data to form decoded image data, comprising:

a command processor that processes the decoding commands, and

a decoding device;

wherein the decoder decodes compressed image data contained in the compressed image data file based on the decoding commands to form a reconstructed image, the reconstructed image representing an altered version of the original image.

3. A data structure for compressed image data file, comprising:

a header; and

a compressed data portion;

wherein the header comprises at least one of:

an identification portion;

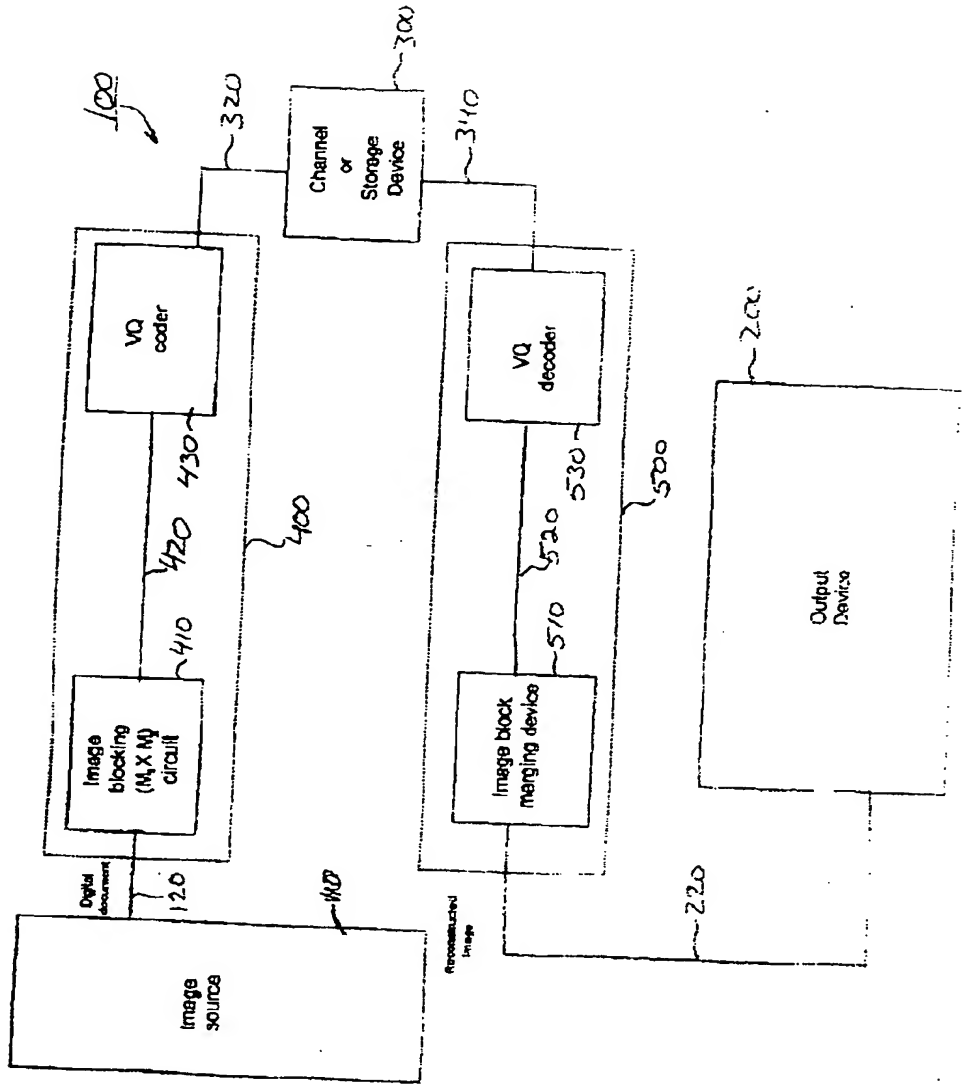
an image information portion;

a photometry portion;

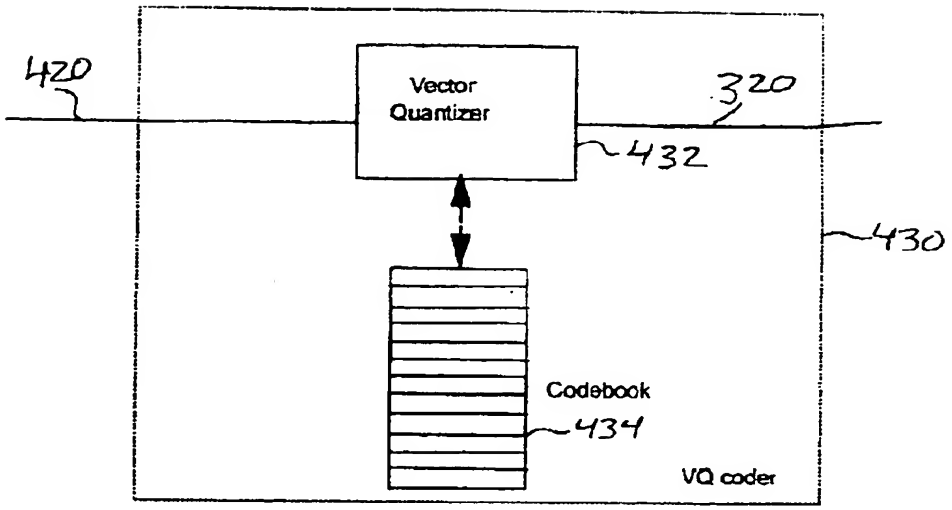
a code portion; and

a codebook definition portion.

【図１】

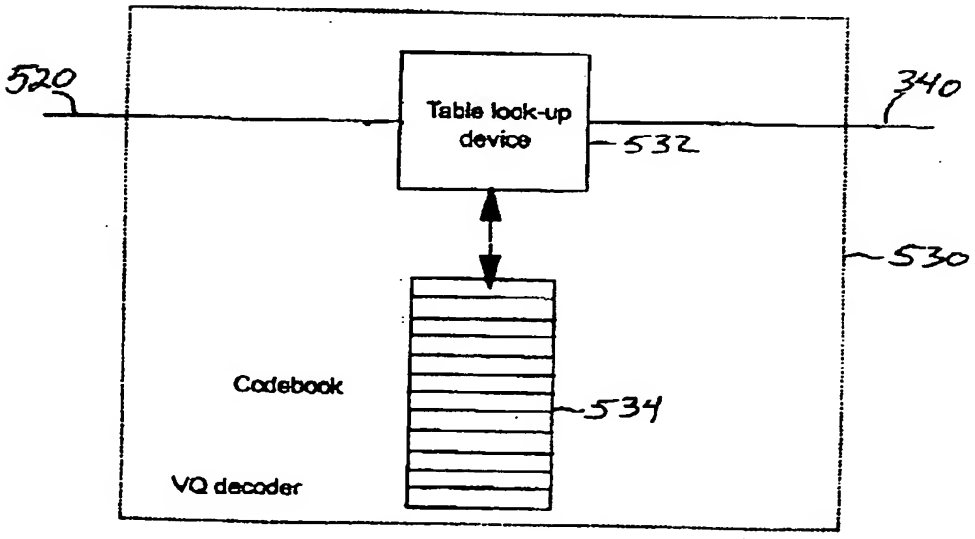

100
320
Channel or Storage Device
310
300
VQ coder
VQ decoder
430
420
530
520
400
500
200
410
510
Image blocking (M_1 X M_2) circuit
Image block merging device
Output Device
Digital document
120
110
220
Image source
Reconstructed Image

【図2】



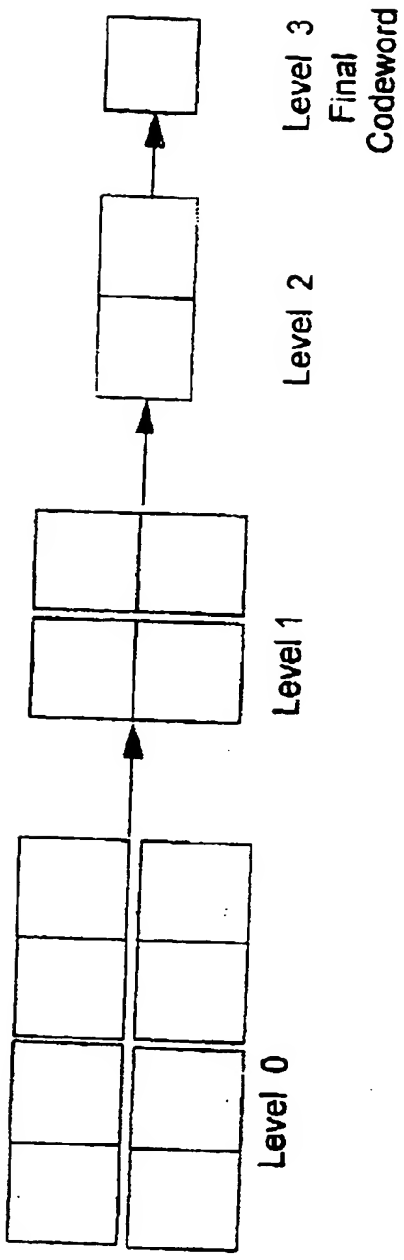
(PRIOR ART)

【図3】

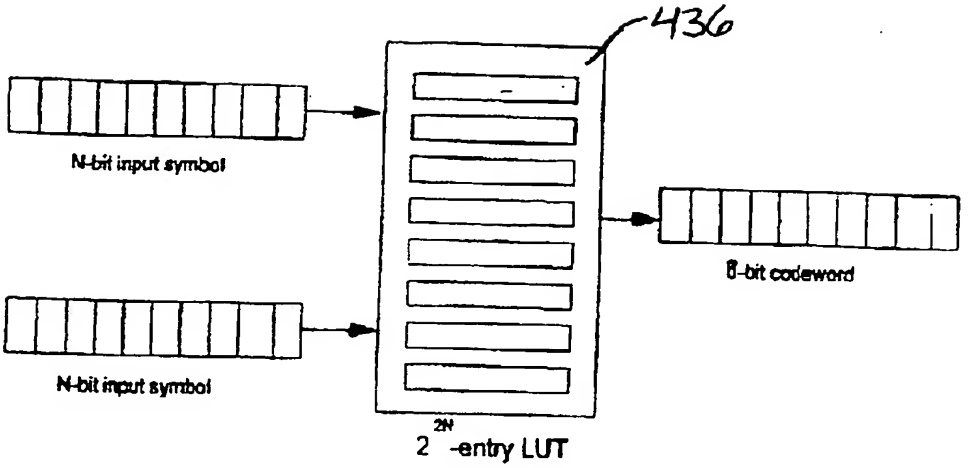


(PRIOR ART)

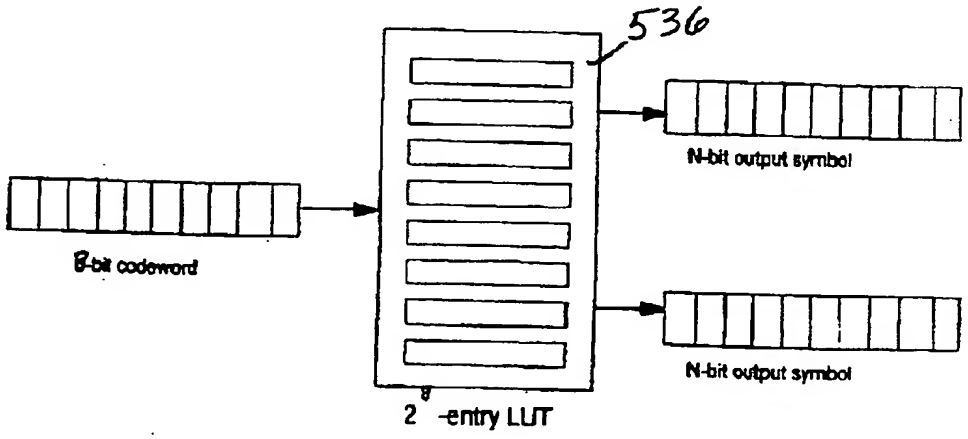
〔図４〕



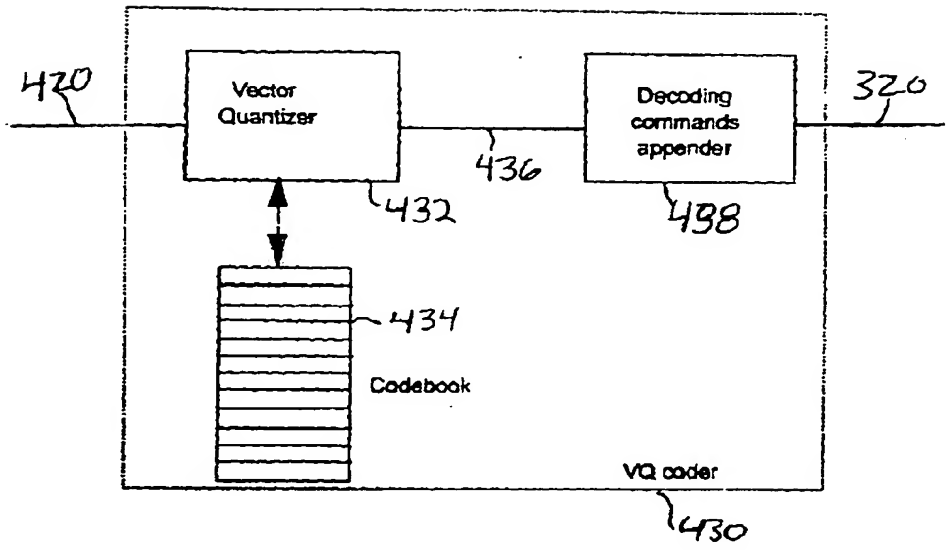
〔図５〕


436
N-bit input symbol
N-bit input symbol
8-bit codeword
2N
2 -entry LUT

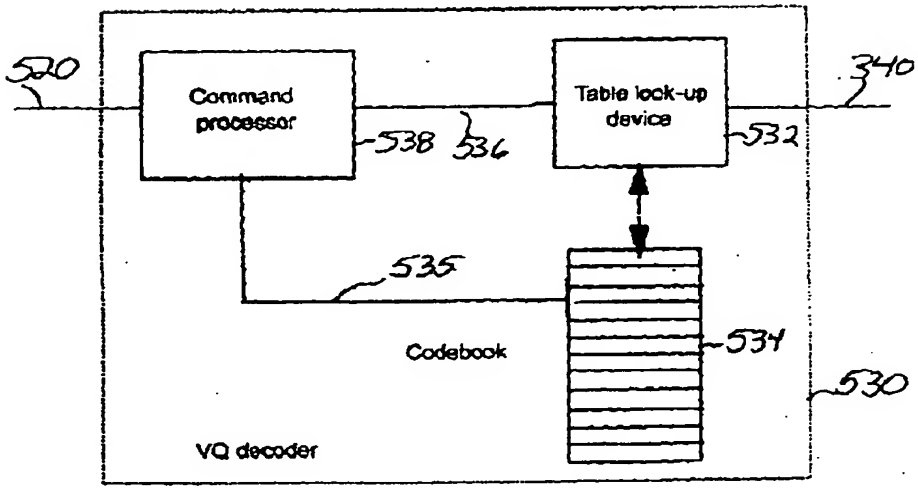
〔図６〕


536
8-bit codeword
N-bit output symbol
N-bit output symbol
8
2 -entry LUT

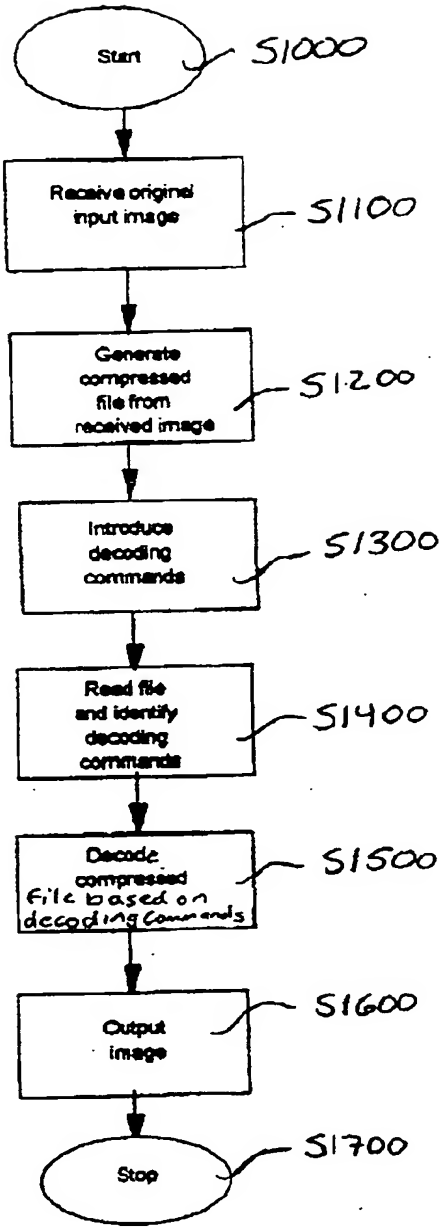
【図７】


420
Vector Quantizer
436
Decoding commands appender
320
432
438
434
Codebook
VQ coder
430

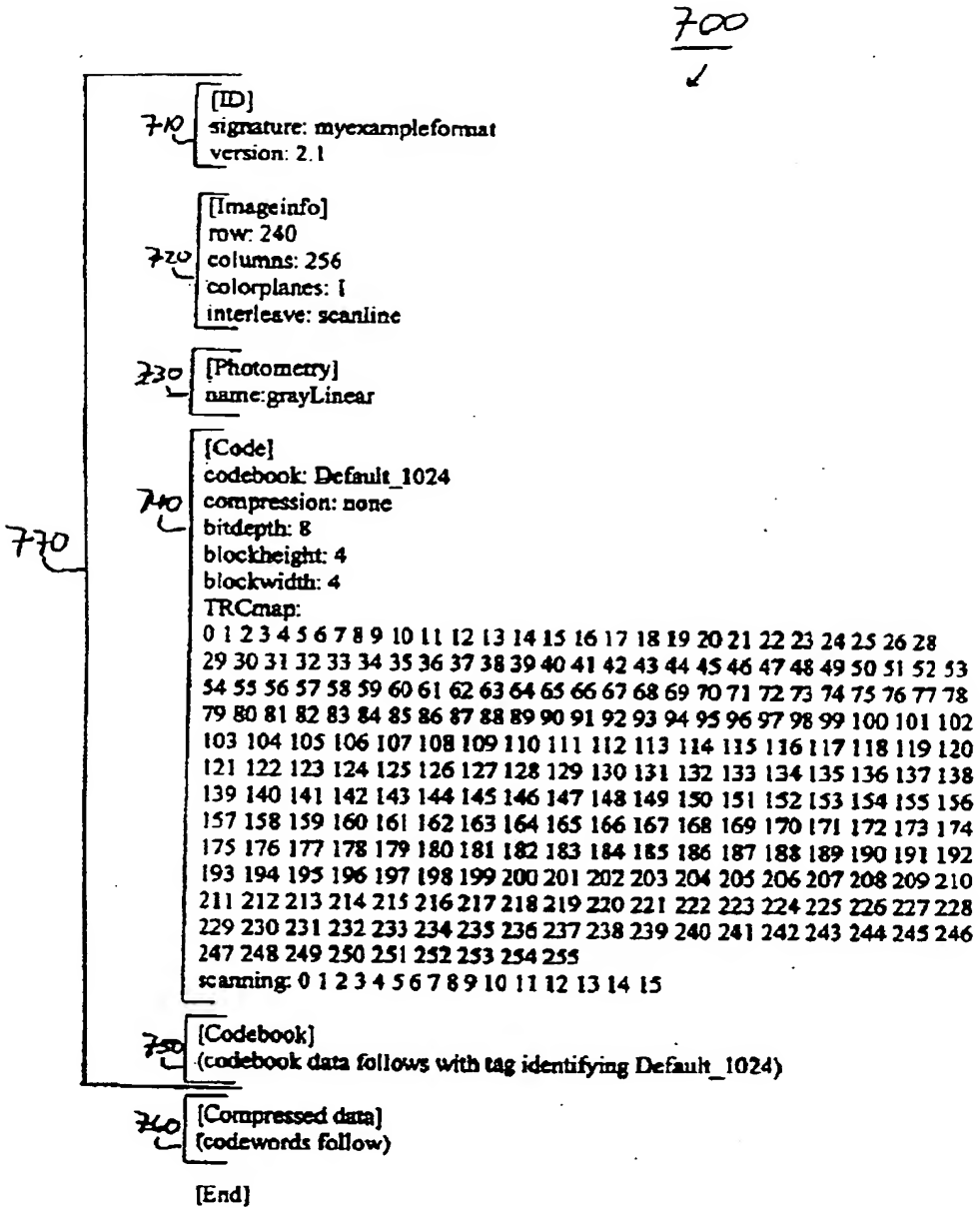
【図８】


520
Command processor
538
536
Table look-up device
532
340
535
Codebook
534
530
VQ decoder

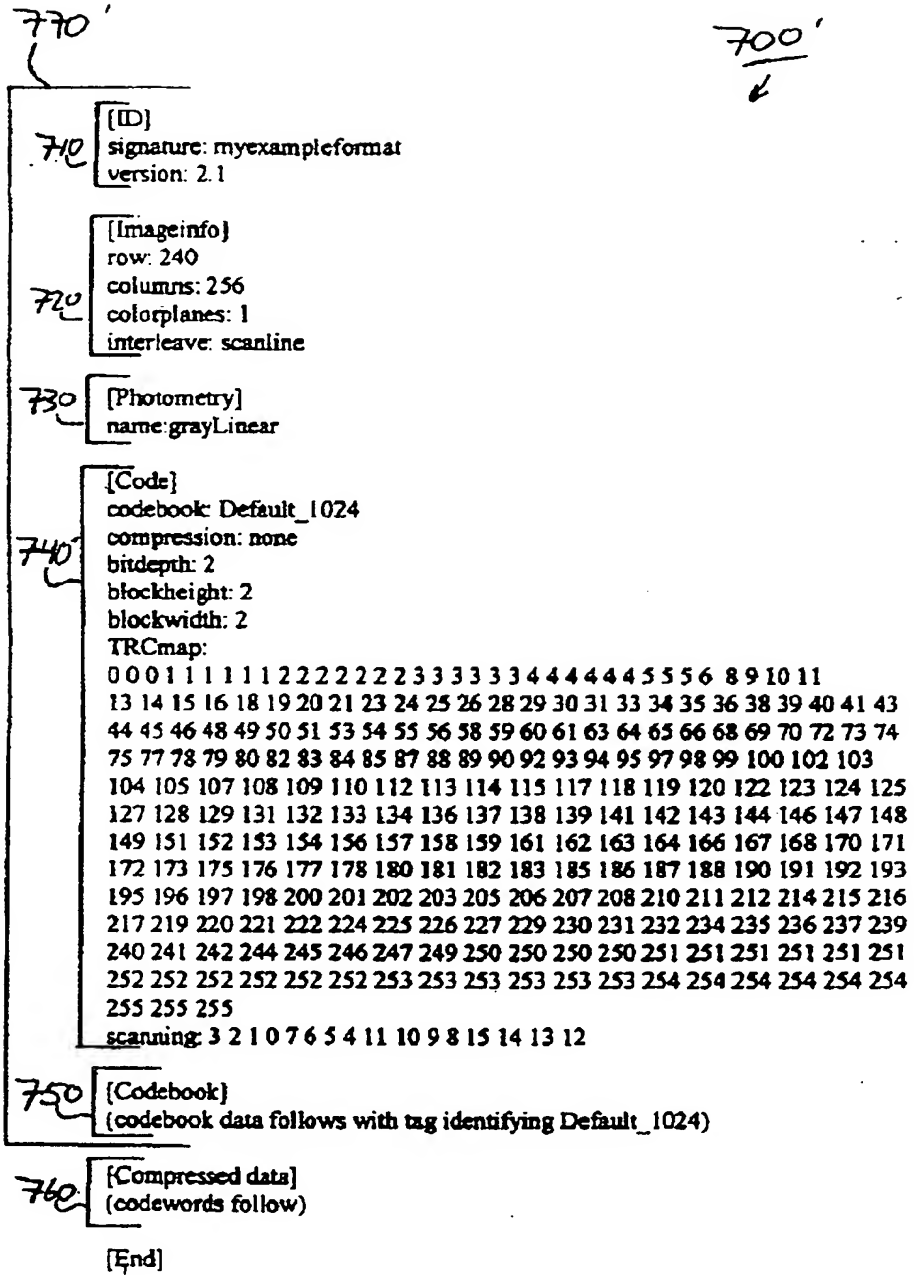
【図9】


Start
S1000
Receive original input image
S1100
Generate compressed file from received image
S1200
Introduce decoding commands
S1300
Read file and identify decoding commands
S1400
Decode compressed file based on decoding Commands
S1500
Output image
S1600
Stop
S1700

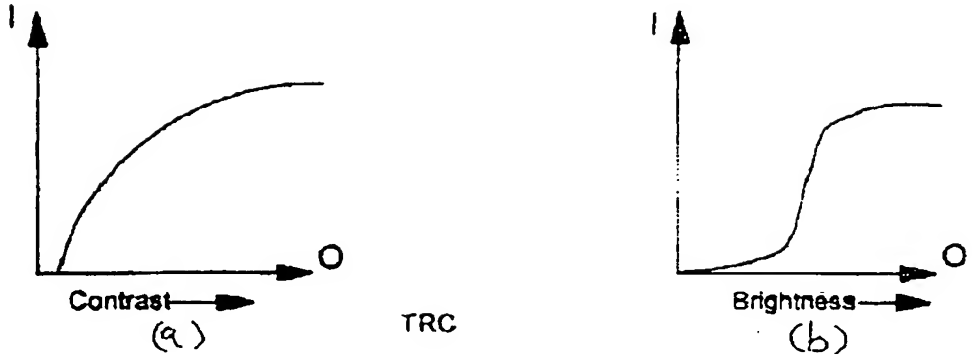
〔図１０〕


700
710
[ID]
signature: myexampleformat
version: 2.1
720
[Imageinfo]
row: 240
columns: 256
colorplanes: 1
interleave: scanline
730
[Photometry]
name:grayLinear
740
[Code]
codebook: Default_1024
compression: none
bitdepth: 8
blockheight: 4
blockwidth: 4
TRCmap:
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138
139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156
157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174
175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192
193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210
211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228
229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246
247 248 249 250 251 252 253 254 255
scanning: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
770
750
[Codebook]
(codebook data follows with tag identifying Default_1024)
760
[Compressed data]
(codewords follow)
[End]

[図11]


770'
700'
710
[ID]
signature: myexampleformat
version: 2.1
720
[Imageinfo]
row: 240
columns: 256
colorplanes: 1
interleave: scanline
730
[Photometry]
name:grayLinear
740
[Code]
codebook: Default_1024
compression: none
bitdepth: 2
blockheight: 2
blockwidth: 2
TRCmap:
0 0 0 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 5 5 5 6 8 9 10 11
13 14 15 16 18 19 20 21 23 24 25 26 28 29 30 31 33 34 35 36 38 39 40 41 43
44 45 46 48 49 50 51 53 54 55 56 58 59 60 61 63 64 65 66 68 69 70 72 73 74
75 77 78 79 80 82 83 84 85 87 88 89 90 92 93 94 95 97 98 99 100 102 103
104 105 107 108 109 110 112 113 114 115 117 118 119 120 122 123 124 125
127 128 129 131 132 133 134 136 137 138 139 141 142 143 144 146 147 148
149 151 152 153 154 156 157 158 159 161 162 163 164 166 167 168 170 171
172 173 175 176 177 178 180 181 182 183 185 186 187 188 190 191 192 193
195 196 197 198 200 201 202 203 205 206 207 208 210 211 212 214 215 216
217 219 220 221 222 224 225 226 227 229 230 231 232 234 235 236 237 239
240 241 242 244 245 246 247 249 250 250 250 250 251 251 251 251 251 251
252 252 252 252 252 252 253 253 253 253 253 253 254 254 254 254 254 254
255 255 255
scanning: 3 2 1 0 7 6 5 4 11 10 9 8 15 14 13 12
750
[Codebook]
(codebook data follows with tag identifying Default_1024)
760
[Compressed data]
(codewords follow)
[End]

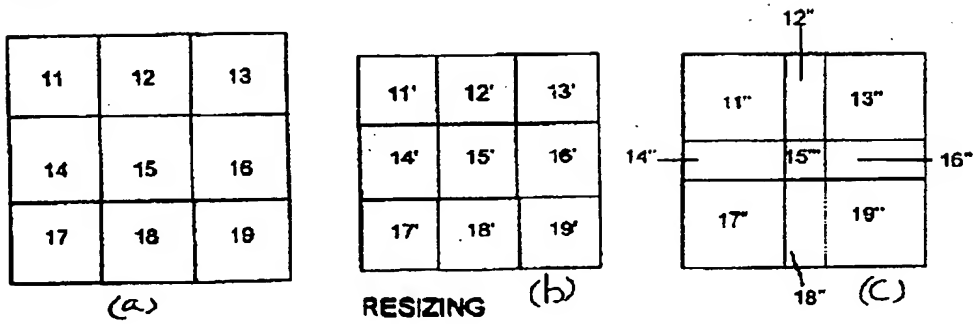
【図１２】


I
O
Contrast
(a)
TRC
I
O
Brightness
(b)

【図１３】


LSB
MSB
(a)
BIT-DEPTH
CONVERSION
MSB
(b)

【図１４】


11
12
13
14
15
16
17
18
19
(a)
11'
12'
13'
14'
15'
16'
17'
18'
19'
RESIZING
(b)
12"
11"
13"
14"
15"
16"
17"
19"
18"
(C)

[図15]

11	12	13
14	15	16
17	18	19

Original
(a)

13'	16'	19'
12'	15'	18'
11'	14'	17'

Rotate 90°
(b)

[図16]

21	22	23
24	25	26
27	28	29

Original
(a)

27'	28'	29'
24'	25'	26'
21'	22'	23'

Vertical Mirror
(b)

【図17】

31	32	33
34	35	36
37	38	39

Original
(a)

31'	34'	37'
32'	35'	38'
33'	36'	39'

Transpose
(b)

[DOCUMENT NAME] ABSTRACT OF THE DISCLOSURE

[SUMMARY]

[PROBLEM TO BE SOLVED]

To incorporate decoding commands into a compressed image data file to complement processing performed in the compressed domain to reduce computation time for imaging operations.

[MEANS TO SOLVE THE PROBLEM]

A method for processing compressed image data, comprising: receiving electronic image data representing an original image (S1100); generating a compressed image data file from the electronic image data (S1200); incorporating decoding commands into the compressed image data file, the decoding commands comprising instructions which alter an appearance of a decoded image relative to the original image (S1300); identifying the decoding commands (S1400); and decoding compressed image data contained in the compressed image data file according to the identified decoding commands to form processed electronic image data of the decoded image, the decoded image representing an altered version of the original image processed (S1500).

[SELECTED FIGURE]

FIG. 9